BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 6 : G06F 1/00 — A1

(11) Numéro de publication internationale: **WO 99/39256**

(43) Date de publication internationale: 5 août 1999 (05.08.99)

(21) Numéro de la demande internationale: PCT/FR99/00182

(22) Date de dépôt international: 29 janvier 1999 (29.01.99)

(30) Données relatives à la priorité:
98/00961 29 janvier 1998 (29.01.98) FR

(71)(72) Déposant et inventeur: LEE, Chiun-Qiang [FR/FR]; Bâtiment A4 Boîte 15, 11, rue Truillot, F-94200 Ivry sur Seine (FR).

(81) Etats désignés: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, brevet ARIPO (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

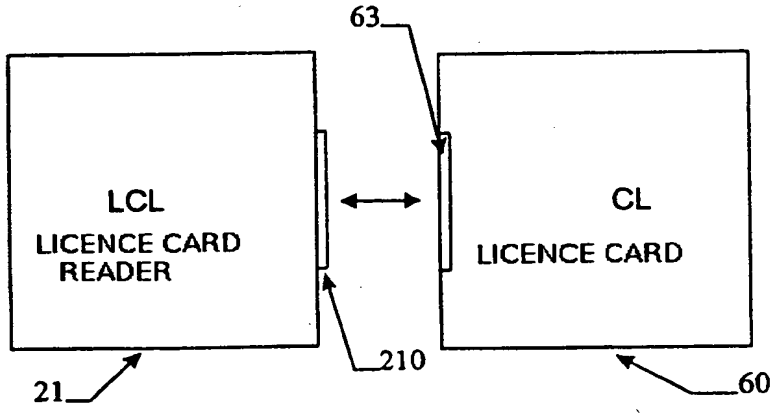
Publiée
Avec rapport de recherche internationale.

(54) Title: SIMULTANEOUS PROTECTION FOR SEVERAL TYPES OF SOFTWARE OF SEVERAL SOFTWARE DESIGNERS

(54) Titre: PROTECTION SIMULTANEE DE PLUSIEURS LOGICIELS DE PLUSIEURS CONCEPTEURS DE LOGICIELS

(57) Abstract

The invention concerns the protection of several types of software against unauthorised use, consisting in an apparatus (licence card reader LCR) for simultaneously protecting several types of software of several software designers and comprising at least one communication peripheral (network, I/S port), a microcontroller programmable only once which integrates on the same silicon chip two parts separated by an interface. The integrated circuit is logically and physically protected against all attempts of unauthorised intrusion. The invention also concerns a portable apparatus of relatively small size with respect to the chip card connected for its use with the LCR apparatus, comprising at least a detachable recording module with high storage capacity, a microcontroller made secure against all attempts at unauthorised intrusion into its internal circuits. Thus the invention provides protection for several types of software independently of their editors with a single apparatus.



(57) Abrégé

La présente invention concerne la protection de logiciels contre leurs utilisations non autorisées. Appareil (lecteur LCL) permettant la protection simultanée de plusieurs logiciels de différents concepteurs de logiciels et comprenant au moins un périphérique de communication (réseau, port E/S), un microcontrôleur programmable une seule fois qui intègre sur une même pastille de silicium deux parties séparées par une interface. Le circuit intégré est protégé logiquement et physiquement contre toutes tentatives d'intrusion non autorisées. Appareil portatif de petite taille par rapport à une carte à puces rattaché dans son utilisation à l'appareil LCL. Il constitue avec l'appareil LCL le deuxième élément de la présente invention. Il comprend au moins un module d'enregistrement amovible de fortes capacités de stockage, un microcontrôleur sécurisé contre toutes intrusions non autorisées dans ses circuits internes. Ainsi, la présente invention réalise la protection de plusieurs logiciels indépendamment de leurs éditeurs par un seul appareil.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	**ES**	Espagne	**LS**	Lesotho	**SI**	Slovénie
AM	Arménie	**FI**	Finlande	**LT**	Lituanie	**SK**	Slovaquie
AT	Autriche	**FR**	France	**LU**	Luxembourg	**SN**	Sénégal
AU	Australie	**GA**	Gabon	**LV**	Lettonie	**SZ**	Swaziland
AZ	Azerbaïdjan	**GB**	Royaume-Uni	**MC**	Monaco	**TD**	Tchad
BA	Bosnie-Herzégovine	**GE**	Géorgie	**MD**	République de Moldova	**TG**	Togo
BB	Barbade	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tadjikistan
BE	Belgique	**GN**	Guinée	**MK**	Ex-République yougoslave de Macédoine	**TM**	Turkménistan
BF	Burkina Faso	**GR**	Grèce			**TR**	Turquie
BG	Bulgarie	**HU**	Hongrie	**ML**	Mali	**TT**	Trinité-et-Tobago
BJ	Bénin	**IE**	Irlande	**MN**	Mongolie	**UA**	Ukraine
BR	Brésil	**IL**	Israël	**MR**	Mauritanie	**UG**	Ouganda
BY	Bélarus	**IS**	Islande	**MW**	Malawi	**US**	Etats-Unis d'Amérique
CA	Canada	**IT**	Italie	**MX**	Mexique	**UZ**	Ouzbékistan
CF	République centrafricaine	**JP**	Japon	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Pays-Bas	**YU**	Yougoslavie
CH	Suisse	**KG**	Kirghizistan	**NO**	Norvège	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	République populaire démocratique de Corée	**NZ**	Nouvelle-Zélande		
CM	Cameroun			**PL**	Pologne		
CN	Chine	**KR**	République de Corée	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Roumanie		
CZ	République tchèque	**LC**	Sainte-Lucie	**RU**	Fédération de Russie		
DE	Allemagne	**LI**	Liechtenstein	**SD**	Soudan		
DK	Danemark	**LK**	Sri Lanka	**SE**	Suède		
EE	Estonie	**LR**	Libéria	**SG**	Singapour		

Protection simultanée de plusieurs logiciels de plusieurs concepteurs de logiciels.

La présente invention concerne la protection de logiciels contre leurs utilisations non autorisées.

L'industrie du logiciel est sans doute le secteur où les produits sont le plus facilement copiés. Or les médias d'enregistrement d'information qui sont en générale des supports optiques, magnétiques sont de plus en plus puissants en capacités de stockages. De plus, le temps mis pour effectuer une copie de ces médias est rapide. De plus, le prix pour posséder un appareil puissant de stockage d'information (disques dures, graveurs de CDROM) a été complètement démocratisé de sorte que les nouvelles versions de logiciels mises en vente sont très rapidement confrontées à des problèmes de copies illicites. De plus, il existe des pays où la copie illicite est pratiquée de manière industrielle et impunément à l'aide de CDROM. Si une telle pratique devait se généraliser, c'est tout le monde informatique qui s'écroule. Les programmes sont développés par les concepteurs de logiciels, appelés aussi développeurs. Les licences d'utilisation de leurs logiciels sont ensuite vendues aux clients. Les sociétés concepteurs de logiciels gagnent des profits généralement à travers la vente directe de leurs produits logiciel et / ou la vente de licences.

L'utilisation illicite de logiciels est définie par rapport à une autorisation d'utilisation de ces logiciels. Cette autorisation se traduit donc par le fait que le concepteur de logiciel accepte de donner une licence d'utilisation qui autorise vis à vis de la loi, l'utilisation de ses produits, au terme d'une entente commerciale.

Le prix de vente des logiciels est calculé avec le nombre d'utilisateurs susceptibles de les acheter. Ainsi les profits dégagés par une société de conception de logiciels dépendent assez de la manière dont leurs clients comptent utiliser ces logiciels une fois achetée. Dans la mesure où après l'achat l'utilisateur est libre de dupliquer le contenu du média contenant ces logiciels, la survie des concepteurs de logiciels dépend assez de l'honnêteté de leurs clients.

Ainsi, pour les logiciels utilisés en réseau, cette licence obtenu autorise en général, l'utilisation du logiciel donné que sur un seul poste d'ordinateur. Pour être utilisé sur plusieurs postes, un nombre de licences correspondant au nombre de postes d'ordinateurs prévus pour l'utilisation de ces logiciels sur son réseau, doit être acheté. Bien entendu pour un ordinateur personnel, ce nombre est égal à 1. Par rapport aux concepteurs, rien ne peut leur garantir qu'effectivement leurs clients respectent bien les conditions liées aux contrats de vente des licences, car en l'absence de méthode et/ou moyen, rien n'empêche l'utilisation du logiciel sur un nombre de poste supérieur au nombre de licences achetées.

De plus, pour les logiciels utilisés sur un ordinateur isolé (ordinateur personnel) ou en réseau, si aucun moyen n'a été prise par le concepteur de logiciels, rien n'empêche à ce qu'un utilisateur pirate face des copies de ces logiciels sur un média informatique pour les installer sur un nombre d'ordinateurs, de manière illimitée et de les utiliser impunément. Il se crée alors des marchés noirs de ventes de logiciels « piratés ». Ce marché non contrôlé peut causer de grand dommage dans l'industrie du logiciel.

Les concepteurs qui désirent contrôler ce phénomène de copies et/ou d'utilisations illicites de leurs logiciels, achètent un appareil électronique permettant de protéger d'une certaine manière ces logiciels. Mais cette solution n'est envisageable que pour certains logiciels. De plus, ces concepteurs sont dépendants du fournisseur de ces moyens de protections de logiciels. Les concepteurs de logiciels à petit budget n'ont pas les moyens de protéger leurs produits numériques compte tenu du prix trop élevé des méthodes de protections par rapport aux prix de vente de leurs logiciels.

De plus, l'utilisation de ces appareils électroniques nécessite de la part du concepteur de logiciels, un achat de ces appareils avant la vente réel de ses logiciels. Cette situation oblige la constitution d'un stock qui peut représenter un désavantage par rapport à ses concurrents qui auraient choisi de ne pas utiliser de moyens de protections de logiciels.

Pour répondre à tous ces problèmes d'utilisation de logiciels, différentes solutions ont été apportées jusqu'ici. Une solution de protection de logiciels en réseaux ou en monopostes (ordinateurs personnels) est proposée dans le brevet U.S. Pat. No. 5,553,139. Cependant il ne permet pas de protéger des logiciels par un même système d'appareil provenant de plusieurs concepteurs différents de logiciels.

D'autres méthodes sont utilisées pour un ordinateur donné par l'intermédiaire de systèmes électroniques connectés directement sur un port E/S de l'ordinateur hôte. Un tel système est proposé dans le brevet U.S. Pat. No. 5,343,524. Cette invention repose sur l'utilisation d'un circuit électronique basé sur un microcontrôleur sécurisé, qui ne peut être reproduit. La protection de logiciels par rapport à cette invention concerne le fait que les logiciels protégés puissent vérifier par l'intermédiaire de clés, la présence de cet appareil et interagir avec cet appareil. Cependant les appareils selon cette invention présentent le désavantage de ne protéger que des logiciels de grandes productions en raison de son coût, et d'autre part de ne pouvoir protéger que les logiciels d'un même concepteur. Un exemple de produit similaire est distribué par la société Rainbow.

D'autre part, des méthodes de protection entièrement logiciel sont aussi employées. Ces protections consistent bien souvent à demander un code d'accès à l'utilisateur. Ce code est ensuite vérifié à l'aide d'un calcul très compliqué. Cependant, il n'empêche pas certains utilisateurs de trouver le type de calcul qui est utilisé de sorte que les logiciels protégés de cette manière n'offrent aucune fiabilité par rapport à la protection de logiciels.

Des systèmes plus puissants sont utilisés à travers l'utilisation de coprocesseur pouvant calculer une partie des codes d'un logiciel donné et protégé selon cette méthode. Généralement, ces logiciels ne peuvent être utilisés directement dans l'état actuel où ils ont été livrés à l'utilisateur, car une partie est codée à l'aide de clés de cryptage stockées de manière sécurisée en accès, dans une mémoire de type ROM du coprocesseur. Ce principe implique le fait qu'un utilisateur ayant obtenu une licence d'utilisation d'un logiciel protégé selon cette méthode soit attaché à l'ordinateur hôte sur lequel le logiciel a été installé. De plus, la possibilité de protéger plusieurs logiciels de plusieurs sociétés de concepteur est difficile à mettre en œuvre. Ainsi, un coprocesseur ayant des

caractéristiques similaires est proposé par le brevet U.S. Pat. No. 4,817,140. Le coprocesseur relative à ce brevet est lancé à condition que l'utilisateur dispose d'une clé pour justifier son achat de licences d'utilisation. Un tel système présente un autre désavantage : son utilisation dédiée à la protection de logiciels d'un seul concepteur, peut rendre la présence d'un coprocesseur génant notamment quand d'autres concepteurs décide de fournir un moyen de protections de logiciels similaire. Dans certains cas, l'ajout de nouveau appareil peut être impossible. Par ailleurs, un tel coprocesseur représente un investissement possible qu'avec des logiciels dont le prix est très haut (coûteux) par rapport au prix de ce coprocesseur déjà onéreux. De plus, l'utilisateur est lié à l'ordinateur sur lequel est installé le coprocesseur.

La plupart des systèmes utilisés pour la protection de logiciels sont dédiés à une catégorie précise de logiciels. L'image de ces systèmes vis à vis de l'utilisateur peut être gênant dans la mesure où ils sont assimilés à une sorte de police électronique de surveillance et non de protections. De plus, ces systèmes sont contraignants dans la mesure où l'utilisation de logiciels protégés par ces systèmes est liée à l'ordinateur hôte sur lequel est installé le logiciel. De plus l'écriture d'un logiciel protégé est très dépendante de l'architecture du moyen de protections, ce qui peut rendre le développement du logiciel compliqué. De plus, comme dans le cas des « dongles », l'utilisateur est lié dans l'utilisation des logiciels aux moyens qui servent à la protections de logiciels. Ainsi, à titre d'exemple, si un dongle est perdu, cette perte entraîne très souvent la perte du droit d'utilisation du logiciel attaché au dongle perdu. D'autre part, les systèmes de protection de logiciels ne tiennent pas compte du fait qu'un logiciel attaché à son système électronique de protection peut être volé. Dans ce cas de vol, et d'utilisation illicite par rapport au voleur, il n'y a pas moyen d'empêcher l'utilisation du logiciel volé. De plus, l'utilisateur devra obtenir une nouvelle licence par un nouvel achat.

De plus, certains logiciels protégés sont caractérisés par leur utilisation limitée dans le temps. Un tel système est présenté par le brevet U.S. Pat. No. 4,868,736. Ce brevet à le désavantage de ne pouvoir réaliser que cette fonctionnalité.

Ainsi, la présente invention permet de remédier à tous les inconvénients qui viennent d'être cités.

La présente invention concerne la protection de logiciel contre le non respect des conditions d'utilisations des logiciels fixés par son concepteur. Elle concerne l'utilisation d'un seul appareil pour protéger plusieurs logiciels indépendamment des systèmes informatiques et du concepteur de ces logiciels. Il est basé sur l'utilisation de deux appareils électroniques qui ne peuvent pas être dupliqués sans autorisations. Cette protection contre la duplication des appareils selon la présente invention est réalisée grâce à une méthode d'authentification intégrée dans ces appareils.

Le premier appareil est un lecteur électronique du second. Il est noté LCL pour lecteur de cartes de licences. Ce lecteur assure la quasi-totalité des fonctionnalités de protections de logiciels selon la présente invention.

Le second appareil est une carte électronique, notée CL (carte de licences). Chaque utilisateur qui désire exécuter des logiciels protégés selon la présente invention, doit posséder une carte CL sur lequel les autorisations d'utilisations de logiciels protégés selon la présente invention, sont stockées.

Ainsi, la présente invention sépare sur trois niveaux la protection de logiciels. Dans un premier temps, la présente invention concerne une méthode permettant la séparation du logiciel protégé (media d'enregistrement) du moyen qui réalise la protection de ce logiciel (le lecteur LCL). Dans un deuxième temps, le lecteur LCL est distribué de manière indépendante par rapport à la distribution des logiciels protégés selon la présente invention. Ainsi, le même lecteur LCL, peut être utilisé pour permettre la protection de plusieurs logiciels indépendamment des concepteurs et du nombre de logiciels. L'utilisation de logiciels protégés selon la présente invention, n'est possible que si l'utilisateur dispose de la carte CL qui est distribuée indépendamment du lecteur LCL.

Selon la présente invention, la carte CL est un appareil portatif de petite taille par rapport à une carte à puces. Elle possède un dispositif amovible d'enregistrement de grande capacité. Elle permet de stocker des données de manière sécurisée contre des lectures et/ou modifications non autorisées. Elle est essentiellement utilisée comme un dispositif d'accès permettant l'utilisation des logiciels protégés selon la présente invention. Les conditions d'utilisation d'un logiciel sont fixées par les concepteurs de logiciels. Un utilisateur ne peut exécuter un logiciel qu'à condition de posséder une autorisation qui lui a été fournie sur sa carte CL lors d'une opération d'achat. La carte CL permet de stocker sur un média d'enregistrement amovible un grand nombre d'autorisations d'utilisations de logiciels protégés. Ainsi, la carte CL permet à l'utilisateur de transporter les autorisations d'utilisation de logiciels et de pouvoir utiliser les logiciels correspondants sur tout ordinateur lorsque l'utilisation de ces logiciels dont il a le droit est possible par rapport à la présence ou non de ces logiciels protégés sur cet ordinateur.

La présente invention concerne par rapport à l'usage de cette carte CL un moyen de lutter contre les pertes ou le vol de cette carte CL. En cas de pertes ou de vols, la carte peut être rendue inutilisable par l'organisme qui gère (administre) les appareils selon la présente invention. Une partie importante des licences d'utilisations de ces logiciels peut être récupérer en cas de perte. Ainsi, l'utilisateur ne coure pas le risque de perdre ses droits d'utilisation d'un logiciel lors de pertes de carte CL, ce qui peut être le cas avec les appareils de protection de logiciels dans l'état actuel de l'art.

La présente invention concerne des méthodes de protection de logiciels indépendamment des systèmes informatiques. La présente invention permet la protection de logiciels utilisés en réseau et/ou sur un ordinateur personnel. Le lecteur LCL possède une grande capacité de modularité par rapport aux différents périphériques qui peuvent être ajouté sur son système électronique interne. Ainsi, il très facile de connecter le lecteur LCL sur tout environnement informatique. Ainsi, la protection de logiciels selon la présente invention, est réalisable avec le même lecteur LCL dans

des systèmes informatiques très hétérogènes où de nombreux systèmes différents cohabitent. Le fonctionnement des lecteurs LCL est indépendant de ces systèmes informatiques. Par conséquent, les lecteurs LCL permettent la protection des logiciels indépendamment des systèmes informatiques prévus pour exécuter ces logiciels.

La présente invention permet le développement de logiciels protégés indépendamment des caractéristiques techniques des appareils selon la présente invention. Le développement des logiciels protégés selon la présente invention, est indépendant du fonctionnement interne du LCL. La réalisation d'un logiciel protégé selon la présente invention est rendue possible en faisant exécuter une partie des fonctions qui composent ce logiciel par les ressources interne du LCL. L'écriture de ces fonctions est complètement transparente dans la mesure où le concepteur de logiciel n'est pas tenu de respecter l'architecture électronique du LCL. Le bon fonctionnement de ces fonctions peut même être testé à l'extérieur du lecteur LCL, de sorte que le travail de protections de logiciels pour le concepteur s'arrête à l'écriture de ces fonctions. Ces fonctions sont essentiellement des fonctions de calculs de petite taille par rapport à la taille d'un logiciel standard et dont l'exécution est très rapide. Ainsi, plusieurs logiciels protégés différents peuvent être utilisés dans le cadre d'un réseau avec le même lecteur LCL. De plus, plusieurs logiciels protégés selon la présente invention peuvent donc être utilisés sur un ordinateur personnel avec un seul lecteur LCL. Le fonctionnement du lecteur LCL utilisé avec un ordinateur personnel par rapport à une utilisation dans un environnement réseau ne diffère qu'au niveau des périphériques de communication utilisés dans chaque cas.

Selon la présente invention, le lecteur LCL réalise la protection de logiciel en effectuant des mesures sur l'utilisation de tous les logiciels en cours d'exécution. Le lecteur LCL est selon la présente invention, capable de connaître le nombre de licences utilisées sur un ordinateur et/ou sur tout le réseau auquel il est connecté. Il est capable de connaître la durée d'utilisation d'un logiciel donné par un utilisateur donné. Selon la présente invention, il est capable de connaître toutes les informations d'utilisation concernant un logiciel donnée par rapport au temps. Ces moyens de mesure propre au lecteur LCL permettent au lecteur LCL d'arbitrer l'utilisation des logiciels protégés selon la présente invention. Cet arbitrage est effectué par rapport aux conditions d'utilisation de chaque logiciel protégé selon la présente invention. Ces conditions sont fixées par le concepteur de ces logiciels.

Ainsi, selon la présente invention, la carte CL sert à stocker des autorisations d'utilisations de logiciels et le profil de l'utilisateur par rapport à son utilisation des logiciels protégés selon la présente invention. Le lecteur LCL est chargé de vérifier si le profil de l'utilisateur par rapports à ses droits d'utilisations des logiciels, correspond bien aux conditions d'utilisation fixées par le concepteur de ces logiciels. Ces conditions peuvent être selon la réalisation de la présente invention, des conditions d'utilisations limitées dans le temps, des conditions d'ouverture simultanée d'un nombre de sessions d'exécution limitées par le nombre de licences d'utilisation possédées par le propriétaire de la carte CL.

Ainsi, un seul lecteur LCL peut arbitrer sur un ou plusieurs ordinateurs l'utilisation de un ou plusieurs logiciels différents protégés selon la présente invention. Les règles d'arbitrages peuvent être spécifiques à chaque version de logiciels, ce qui permet la protection de plusieurs logiciels par un seul appareil selon des critères spécifiques à l'utilisation de chaque logiciel protégé.

La présente invention permet une distribution des logiciels protégés indépendamment des moyens de protection de sorte que le développeur de logiciels peut ne pas constituer de stocks d'appareils permettant la protection. Ainsi, la protection de logiciels selon la présente invention est intéressante pour les petites et les grandes distributions de logiciels : l'usage d'un seul appareil pour la protection de plusieurs logiciels indépendamment des concepteurs ne peut que diminuer le coût d'utilisation du système de protection de logiciels selon la présente invention. De plus, la présente invention permet une grande souplesse par rapport à l'administration des logiciels protégés. La vente des autorisations d'utilisation d'un logiciel données peut être centralisée ou décentralisée.

De plus, selon la présente invention, la création de logiciel protégé peut être effectuer de plusieurs manière. Les autorisations pour créer un logiciel protégé selon la présente invention, peuvent être centralisées ou décentralisé (situation décrite dans le cas de développement de logiciels de démonstrations ou de logiciels dont l'utilisation est limitée).

De plus, les appareils selon la présente invention peuvent effectuer des communications avec des systèmes distants. L'administration de ces appareils est effectuée par un système distant qui est selon la réalisation de la présente invention, un serveur noté aSVR. Ce serveur fixe les conditions d'utilisations des appareils selon la présente invention. Il arbitre de manière générale l'utilisation des appareils selon la présente invention.

De plus, les autorisations d'utilisation de logiciels contenue dans une carte CL peuvent être soit déplacées dans le lecteur LCL, soit dans une autre carte CL. Dans le cas où le déplacement aurait lieu vers un lecteur LCL, l'accès aux logiciels protégés peut être réalisé sans la présence d'une carte CL. Dans le cas de déplacements vers une autre carte CL, cela permet des distributions de licences d'utilisations de logiciels par des revendeurs. La présente invention permet ou non la centralisation des ventes des autorisations de logiciels.

De plus, selon la présente invention, le lecteur de cartes LCL possède un ou des dispositif(s) qui permet un rajout de périphériques rapidement et facilement. Selon des modes particuliers de réalisation, le lecteur LCL dispose d'un récepteur radio pour recevoir des informations de manière sécurisée ou non grâce à un dispositif d'émission géré par ledit serveur aSVR. Ce récepteur est essentiellement utilisé pour des opérations d'achat d'autorisations d'utilisation de logiciels hors ligne, des opérations de mise à jour. Il permet aussi de gérer la sécurité d'utilisation des appareils selon la présente invention.

La présente invention permet par ses moyens et ses méthodes des achats soit par des connexions informatiques, soit par un système d'accueil humain. Les achats peuvent donc être effectués en ligne ou hors ligne. Ces achats consiste en l'acquisition des autorisations d'utilisations

de logiciels protégés selon la présente invention. La présente concerne l'emploi d'un récepteur radio numérique permettant en particulier la réception de ces autorisations d'utilisations.

Pour réaliser la protection de plusieurs logiciels indépendamment de leurs concepteurs, la présente invention concerne l'utilisation au sein du lecteur LCL, d'un microcontrôleur sécurisé contre les lectures et/ou les modifications non autorisées de sa mémoire interne et contre des attaques de virus informatiques qui peuvent être rencontrés dans la mesure où ce microcontrôleur exécute des programmes dont il ignore la fiabilité, l'utilisation du lecteur comme moyen de protection de logiciels est libre en dehors des accords commerciaux éventuels. Ainsi, cette propriété permet à ce que l'usage du lecteur LCL et de la carte CL sont complètement libre.

Pour permettre à l'utilisateur de transporter des autorisations d'utilisations de logiciels protégés, la carte CL est de petite taille. Il est basé sur un microcontrôleur permettant de sécuriser l'accès aux informations qui définissent les droits d'utilisations d'un logiciel donné. Il peut stocker un très grand nombre d'autorisation d'utilisations de logiciels protégés selon la présente invention. Les informations sont stockées sur un média d'enregistrement de fortes capacités. Elles sont protégées contre toutes modifications et lectures non autorisées. De plus, le système interne de son microcontrôleur est aussi protégé contre tout contrôle physique et logique.

De plus, la présente invention concerne un système de protection de logiciel évolutif dans la mesure où il est possible de mettre à jour l'ensemble des systèmes informatiques contenus dans les microcontrôleurs cités précédemment. Compte tenu de leur capacité de sécuriser le stockage de données et l'exécution de programmes, compte tenu aussi de la possibilité de mettre facilement à jour les systèmes informatiques des appareils selon la présente invention, la présente invention permet à des appareils de protection de logiciels d'être utilisée dans d'autres secteurs d'applications.

Les dessins annexés illustrent l'invention :

La figure 1 illustre l'ensemble des variantes de connexion mise en jeux dans la présente invention, et reflète le fonctionnement général de l'invention. Elle permet de comprendre les différents contextes d'utilisations des appareils selon la présente invention.

La figure 2 illustre les différentes couches de logiciels nécessaires pour que un LCL donné puissent se connecter vers un système distant.

La figure 3 illustre le schéma synoptique de l'architecture du microcontrôleur utilisé dans l'appareil LCL.

La figure 4 reflète les associations possibles entre deux LCL concurrents sur le même réseau, pour permettre le partage des opérations liées à la protection de logiciels par la présente invention.

La figure 5 illustre le schéma synoptique des différents éléments composant la carte électronique CL, notamment l'architecture du microcontrôleur associé à une carte CL.

La figure 6 représente une vue de face du boîtier de la carte CL.

La figure 7 représente une vue en perspective du boîtier CL avec la carte CompactFlash sortie de son support.

La figure 8 illustre les jeux de connecteurs mâles femelles entre LCL et CL.

La figure 9 illustre le fait que les appareils selon la présente invention possèdent des durées d'utilisation par rapport au calendrier.

La figure 10 illustre les étapes d'une procédure d'authentification empêchant des appareils pirates de fonctionner avec les appareils (lecteurs LCL et cartes CL) certifiés.

La figure 11 illustre les étapes d'une opération d'achat de licences d'utilisation de logiciels protégés selon la présente invention.

La figure 12 illustre un arbre logique, simplifié et utilisé par le système d'exploitation du microcontrôleur 100 pour effectuer la protection de logiciel.

En référence à la figure 1, selon la réalisation de la présente invention, l'ensemble du système qui exploite le lecteur électronique, peut avoir une gestion centralisée par l'intermédiaire d'un serveur aSVR qui possède une base de données 12 relatives aux appareils selon la présente invention. L'ensemble des éléments compris dans l'encadré 10 est géré par un organisme donné. Cet organisme est le distributeur des appareils relatifs à la présente invention. Le serveur aSVR peut communiquer avec des systèmes informatiques distants utilisant un lecteur LCL. Selon des variantes de la réalisation de la présente invention, un lecteur LCL peut posséder un moyen pour se connecter soit directement sur un réseau 40, soit sur un port E/S d'un ordinateur personnel (contexte représenté par l'encadré 30) ou posséder un radiorécepteur numérique 22 (contexte représenté par l'encadré 20).

Selon la réalisation de la présente invention, la figure 2 illustre les différentes couches traversées par LCL pour atteindre un système distant. Les communications entre le système distant et LCL sont gérées par deux programmes tournant sur un ordinateur 50. La connexion 54 entre l'ordinateur et un lecteur LCL peut en fonction du type d'utilisation être une connexion réseau dans le cas de l'encadré 40 ou une connexion directe sur un port E/S de l'ordinateur 50 dans les contextes 30 et 20. Le port retenu pour la réalisation de la présente invention, dans les contextes 30 et 20, est un port USB (Universal Serial Bus) pour des raisons de rapidité, en supposant que les ordinateurs utilisés possèdent un tel port de communication. La description de la présente invention a retenu pour le contexte d'utilisation 40, le cas d'un réseau Ethernet avec le protocole TCP/IP. Ainsi, dans ce contexte le lecteur LCL possédera un périphérique réseau adéquat. Le programme PGM 52 permet de communiquer de manière interactive avec un lecteur LCL donné. Ces communications se font grâce au programme driver DRV 51. En référence à la figure 2, ce driver 51 réalise toutes les fonctions de communications entre un lecteur LCL et un ordinateur 50 connecté à ce lecteur. Les différentes fonctionnalités de ces deux programmes propres à la gestion des appareils selon la présente invention, seront définie par la suite. Le programme PGM permet d'assurer au lecteur LCL une communication avec un système distant. Pour réaliser cette communication, PGM utilise les ressources de communications 53 de l'ordinateur hôte 50. Le programme DRV assure quant à lui la communication locale entre le lecteur LCL et le programme PGM. Dans le cas 30, il peut s'agir du modem 31 permettant une connexion vers le réseau Internet

auquel est attaché le serveur aSVR. Selon la réalisation de la présente invention, pour le cas 40, la ressource de communication 53 est celle de l'ordinateur 50 par rapport aux ressources du réseau local, permettant un accès vers un système distant via le réseau Internet.

L'utilisation des lecteurs LCL est sous soumise par une condition d'identification lors de sa mise en marche, et par une condition de la connexion effective de la carte CL sur un support du lecteur LCL. Ces deux conditions seront décrites par la suite.

Le contexte 20 correspondant à des situations où le lecteur LCL est connecté sur un ordinateur ne possédant pas de moyens de communication avec un système distant. Ce fonctionnement sera décrit par la suite.

Selon la présente invention, la protection de logiciel est rendue possible en faisant exécuter une petite partie des fonctions d'un logiciel donné, par ledit lecteur électronique, noté LCL. Ladite carte électronique, noté CL, possède les autorisations d'utilisation du logiciel. Compte tenu de sa capacité de stockage, la carte CL peut stocker une grande quantité d'autorisations d'utilisation. De cette manière, la présente invention permet par le moyen d'un seul dispositif électronique, la protection de plusieurs logiciels simultanément et indépendamment de leurs concepteurs.

Selon la réalisation de la présente invention, le lecteur LCL et la carte CL sont respectivement associés à deux numéros de série uniques. L'ensemble des informations permettant le fonctionnement du lecteur LCL et de la carte CL pour réaliser la protection de logiciels, est géré par le serveur aSVR à l'aide de sa base de données 12 qui doit être protégée contre des accès non autorisés par rapport à la sécurité du système mise en place pour la protection de logiciels. L'organisme en question qui gère aSVR attribue respectivement à une carte CL donné et un lecteur LCL donné, les numéros de séries ID.c et ID.d, et les clés secrètes de codages kT.c et kT.d. Les numéros ID.c et ID.d sont uniques. Ces deux numéros et ces deux clés secrètes sont stockés dans de la mémoire non volatile qui se trouve dans le système électronique de CL et de LCL. Des précisions seront données ultérieurement. Les couples (ID.c, kT.c) et (ID.d, kT.d) sont stockés par ailleurs dans la base de données 12 accessible uniquement par aSVR pour des raisons de sécurités évidentes. Les clés kT.c et kT.d sont connues uniquement par aSVR, en dehors des appareils LCL et CL.

De plus, selon la réalisation de la présente invention, les numéros ID.c et ID.d sont publics, mais non modifiables. Cela signifie qu'ils sont stockés dans la mémoire protégée du système électronique intégré dans chaque appareil relatif à la présente invention, et qu'ils sont communiqués par ailleurs à leur utilisateur sous une forme claire. Selon la réalisation de la présente invention, ils sont marqués sur le boîtier de CL et de LCL. De plus, la présente invention ne concerne la forme du boîtier utilisé pour le lecteur LCL.

Selon la réalisation de la présente invention, la méthode de cryptage adoptée avec les clés secrètes de codage kT.d et kT.c, concerne le cryptage DES (data encryption standard) développé par la société IBM. Ces deux clés ont une taille de 128 bits suffisante contre des craquages de codes. Selon des modes particuliers de réalisation, on pourra choisir d'autre type de cryptage et de

taille de clés de codage.

Selon la présente invention, les appareils LCL et CL sont paramétrables dans une certaine mesure par l'intermédiaire d'un programme informatique, noté PGM, adapté pour chaque type d'ordinateur et de systèmes d'exploitation informatique. PGM permet à un utilisateur de démarrer des procédures relatives à des opérations nécessitant une intervention de l'utilisation. Ces procédures sont décrites par la suite. PGM est distribué par ledit organisme qui gère aSVR. A l'installation de PGM sur un ordinateur hôte, une opération de localisation d'un lecteur LCL est effectuée. Si un lecteur LCL est connecté directement sur un port de communication de l'ordinateur hôte, un driver logiciel DRV de communication à travers ce port est installé afin de permettre à un programme de cet ordinateur d'envoyer des données vers LCL et d'en recevoir de LCL en suivant le schéma de la figure 2, sans tenir compte des caractéristiques techniques de la communication entre cet ordinateur et le lecteur LCL connecté. Le driver DRV permet une utilisation transparente du lecteur LCL.

Dans le cas où des logiciels protégés seraient utilisés en réseau (il s'agit du contexte 40), un programme driver DRV adéquat sera installé sur chaque ordinateur du réseau, pour permettre la communication entre ces ordinateurs et le LCL connecté sur ce réseau local. Le driver DRV permet une utilisation transparente de LCL pour chaque ordinateur du réseau qui peuvent avoir des systèmes informatiques différents entre eux.

A son installation sur le réseau Ethernet, le lecteur LCL reçoit une adresse IP par rapport au protocole TCP/IP, qui permet au driver installé sur les ordinateurs du réseau de le localiser.

De plus, chaque driver DRV permet à chaque ordinateur hôte où sont utilisés des logiciels protégés par la présente invention, de communiquer avec le lecteur LCL. La réalisation de la présente invention permet aussi à ce que plusieurs programme PGM puissent établir des communications avec un lecteur LCL donné connecté sur le réseau considéré. Dans son utilisation réseau 40, le lecteur LCL en cas d'utilisation intensive, peut partager avec d'autre lecteur LCL, sa fonction de protection de logiciels comme on le voit sur la figure 4 où une répartition des ordinateurs utilisant deux lecteurs LCL est illustrée. La répartition des lecteurs LCL présents sur le réseau avec les ordinateurs est, selon la réalisation de la présente invention, réalisée par l'administrateur réseau. Ce dernier effectue la répartition des ordinateurs du réseau par rapport aux lecteurs LCL présents sur le réseau, au moment de l'installation de DRV sur chaque ordinateur de ce réseau. Il indique à DRV l'adresse IP du lecteur LCL à utiliser.

Après l'installation de PGM sur un ordinateur hôte, les utilisateurs de cet ordinateur peuvent alors effectuer sur un lecteur LCL donné, connecté avec une carte CL donnée les différentes opérations suivantes : achat en ligne de licences d'utilisation de logiciels protégés par la présente invention, déplacement d'un certain nombre de licences d'utilisation de logiciels d'une carte CL vers une autre carte CL, des opérations de récupération de licences perdues avec la perte d'une carte CL, la mise à jour des programmes contenus dans les mémoires électroniques du lecteur LCL ou de la carte CL.

Ainsi, la réalisation de la présente invention considère un logiciel donné, noté LD. Les descriptions suivantes sont valables pour tout autre logiciel. Son concepteur (fabricant) le protège selon la présente invention, contre des utilisations illégales en séparant l'ensemble des fonctions composant son logiciel en deux parties. La première partie concerne les procédures dont il désire laisser l'exécution à un ordinateur hôte. La deuxième partie concerne les fonctions qui devront être exécutées par les ressources de calculs de LCL. Ces fonctions doivent être rapide à exécuter. Leur taille est de l'ordre du 100 kilooctets.

De cette deuxième partie dudit logiciel LD, il en extrait une liste de fonctions $\{F_0, F_1,\ldots,F_i,\ldots F_n\}$. Ces fonctions sont nécessaires pour le fonctionnement du logiciel LD. Pour effectuer cette extraction, il doit respecter une règle primordiale : ces fonctions ne doivent pas faire appel à des ressources caractéristiques des ordinateurs prévus pour l'exécution de LD. Cette condition est assez facile à respecter. Par exemple, ladite liste de fonctions peut être uniquement des fonctions de calculs purs.

Selon la réalisation de la présente invention, l'écriture de ces fonctions et le test de leur bon fonctionnement peuvent être réalisés indépendamment de la présence du lecteur LCL. Le moyen retenu pour la réalisation de la présente invention, concerne la machine virtuelle JAVA. Ainsi, ces fonctions sont écrites en JAVA. Ces fonctions sont donc compilées en « byte code » du langage universel JAVA et stockées dans un fichier, noté LF.

Par ailleurs, selon la réalisation de la présente invention, la protection du logiciel LD commence alors par l'exécution du programme PGM sur un ordinateur contenant LF. Selon la réalisation de la présente invention, PGM demande alors au concepteur de LD le système d'exploitation (WINDOWS NT, DOS, UNIX,...) et le type d'ordinateur (MACINTOSH, SPARC, PC,...) qui exécuteront LD. Après ces réponses, PGM lance une procédure de connexion du lecteur LCL disponible vers le serveur aSVR. La communication entre LCL et aSVR se fait selon la figure 2. Le numéro de série du lecteur LCL est donné en premier au serveur aSVR. Durant cette procédure de création d'un logiciel protégé par la présente invention, LCL demande à aSVR en communication sécurisée un numéro de série S# à associer au logiciel à protéger LD et une clé de codages kX.S#.

Selon la réalisation de la présente invention, S# est définit sur 128 bits, et la clé kX.S# est définie par rapport au cryptage DES avec une taille de 128 bits.

Selon la réalisation de la présente invention, pour effectuer ladite communication sécurisée avec aSVR, LCL commence par envoyer à aSVR son numéro ID.d sous une forme non codée. Par association avec la clé correspondante, aSVR trouve dans sa base de données 12, la clé kT.d à associer à ID.d.

Ainsi, aSVR retourne à LCL, S# et kX.S# sous une forme codée avec la clé kT.d. Le lecteur LCL récupère la forme claire de S# et kX.S# en décodant avec la clé kT.d qui se trouve dans sa mémoire interne 111, par rapport à l'algorithme de cryptage DES.

Selon la présente invention, la clé kX.S# est connue par aSVR seul, car après utilisation,

kX.S# sera effacé de la mémoire sécurisée DRAM 109 de LCL. De plus, S# est communiqué à PGM qui l'inscrit dans un fichier binaire contenant les conditions limites d'utilisation du logiciel. Selon la réalisation de la présente invention, ce fichier peut avoir le format suivant : S# du logiciel associé (128 bits), licences permanentes (8 bits), durée d'utilisation (24 bits), l'utilisation expire fin (16 bits), nombre d'exécutions (32 bits). Ce fichier fait donc une taille de 26 octets. $\{F_0, F_1,\ldots,F_i, \ldots F_n\}$ subissent ensuite des opérations de cryptage. La fonction F_0 est traitée à part. Il s'agit selon la réalisation de la présente invention, de la procédure dite d'initialisation permettant d'une part de compter le nombre de licences utilisées sur le réseau dans le contexte réseau 40, et d'autre part de mesurer des caractéristiques d'utilisation du logiciel LD par rapport au temps. C'est une fonction qui est exécutée durant l'exécution du logiciel LD par un utilisateur, et de façon répétitive. F_0 est en particulier la première fonction qui sera exécutée par LCL lors de l'ouverture d'une session d'exécutions du logiciel LD.

F_0 est codée avec une clé différente de kX.S#. Pour cela le logiciel PGM génère une clé du même type notée kEL.S# par rapport au cryptage DES. La clé kEL.S# est connue uniquement par le concepteur de LD. Selon la réalisation de la présente invention, kEL.S# est une clé de 128bits. Il est de la responsabilité du concepteur de conserver en sécurité cette clé kEL.S#. A cette fonction codée, il joint une autre information codée aussi par la clé kEL.S#. Il s'agit du fichier contenant les conditions limites d'utilisation du logiciel LD. Ces informations codées constituent alors un fichier nommé eF_0 selon la réalisation de la présente invention.

De plus, selon la réalisation de la présente invention, les autres fonctions sont ensuite codées par la clé kX.S#. Pour cette étape, PGM envoie ensuite $F_1,\ldots,F_i, \ldots F_n$ vers LCL par l'intermédiaire de DRV. Les ressources de calculs de LCL codent alors ces fonctions successivement avec la clé kX.S#. A la fin des opérations de codage, LCL retourne $\{eF_1,\ldots,eF_n\}$ correspondant respectivement aux formes codées de $\{F_1,\ldots, F_n\}$.

Selon la réalisation de la présente invention, PGM procède ensuite à l'assemblage du logiciel. Pour un système d'exploitation donnée, PGM crée un fichier bibliothèque de fonctions qui seront exécutée durant l'exécution du logiciel LD sur l'ordinateur d'un utilisateur donné. Les différentes fonctions de la bibliothèque permettent de charger un élément de $\{eF_0, \ldots,eF_n\}$ avec les paramètres nécessaires à l'exécution de la fonction correspondante, vers le lecteur LCL lors de l'utilisation de LD. eF0, ...,eFn seront respectivement chargées par les fonctions $FF_0,\ldots,FF_n$ créées par PGM. Ces fonctions sont créées par rapport au type de systèmes d'exploitations et le type d'ordinateurs associés qui exécuteront LD. PGM rassemble ainsi {FF0,...,FFn} et {eF0,...,eFn} avec le reste du logiciel LD ainsi protégé, et numéroté S#. Le tout est ensuite mis sur média d'enregistrement, par exemple un CDROM. Le logiciel est ainsi protégé et prêt à être diffusé librement, car il ne pourra pas être utilisé dans l'état actuel. Ainsi, la distribution du média d'enregistrement peut être effectuée de manière complètement libre.

Ainsi, la protection de logiciel selon la présente invention, est basée sur l'utilisation des ressources de calculs du lecteur LCL.

Selon la présente invention, le lecteur LCL est un lecteur électronique construit autour d'un microcontrôleur 100 sécurisé physiquement et logiquement afin de prévenir contre les tentatives de pirates pour des contrôles électroniques non autorisés. Compte tenu du fait que ce microcontrôleur 100 est amené à exécuter des programmes d'origine inconnue, la présente invention concerne un moyen d'empêcher des attaques informatiques du microcontrôleur 100 par l'intermédiaire de virus informatiques. Cette mesure est prise pour empêcher un virus de lire des informations confidentielles liées aux fonctionnements du lecteur LCL. Ainsi la présente invention permet de sécuriser à la fois le stockage d'information et à la fois l'exécution de tout programme extérieur aux programmes initialement chargés dans le microcontrôleur 100.

Selon la réalisation de la présente invention, l'architecture retenue du microcontrôleur est construite autour d'un système basé sur un jeu de deux processeurs utilisés en maître esclave. En référence à la figure 3, le microcontrôleur 100 intègre sur la même pastille de silicium deux parties principales 130 et 120. Les méthodes d'intégration ASIC (Application Specific Integrated Circuit) sont employées pour réaliser ces pastilles de silicium. La partie 130 comporte un processeur CPU1 qui est le processeur maître. Il est relié par un bus interne 101 à un module de mémoires FLASH 111, un module de mémoires DRAM 109, un générateur de nombre aléatoire 112, un port E/S RS232 151, un port USB 152, un contrôleur de cartes à puces (SmartCard) 153, un contrôleur PCMCIA 154, un contrôleur clavier et écran LCD 155, un contrôleur 113 du processeur esclave CPU2 qui se trouve dans la partie 120, une interface 106, une interface Bus externe 105, une horloge temps réel programmable interne 104, un système de microfusible interne 102 qui permet de sortir le bus interne 101 à l'extérieur du microcontrôleur 100. La partie 120 du microcontrôleur 100 comporte un watchdog 108. Le CPU2 est relié par le bus interne 114 à un module de mémoires DRAM 110, un contrôleur DMA 107, et une interface 106. L'esclave CPU2 est commandé par l'intermédiaire du contrôleur 113 et de l'interface 106. Ces deux derniers systèmes électroniques (113 et 106) sont contrôlés uniquement par le processeur maître CPU1. Ainsi, une telle architecture permet l'exécution des programmes d'origine inconnue sans pour autant endommager l'intégrité des informations contenues dans le microcontrôleur 100.

Selon des modes particuliers de réalisation, le microcontrôleur 100 peut ne pas intégrer sur la même pastille de silicium tout ou partie des éléments suivants : Port E/S RS232 151, port USB 152, contrôleur de cartes à puces 153, contrôleur PCMCIA 154, contrôleur clavier et écran LCD 155. Selon des modes particuliers de réalisation non illustrés, et afin d'accélérer la vitesse de traitement d'informations, le microcontrôleur 100 peut comporter sur la même pastille de silicium un coprocesseur de cryptage adapté par rapport à la technique de cryptage DES. Bien entendu ce coprocesseur de cryptage sera intégré dans la partie 130 relié sur le bus interne 101.

Selon la présente invention, l'horloge temps réel interne 104 est alimentée par une pile électrique 103 externe par rapport au microcontrôleur 100. Son fonctionnement est autonome. Cette horloge est intégrée sur ladite pastille de silicium afin d'empêcher des tentatives de contrôles électroniques faussant l'heure et la date qu'il fournit au CPU1. Compte tenu de la faible

consommation électrique de cette horloge, la pile électrique permet de fournir en continue le courant nécessaire au fonctionnement de cette horloge pendant toute la durée d'utilisation du microcontrôleur 100 comme organe central du lecteur LCL. Eventuellement, une procédure de mise à l'heure de l'horloge 104 pourra être effectuée par le serveur aSVR. L'horloge 104 permet de mesurer le temps d'utilisation des logiciels protégés et d'effectuer des opérations dépendant de la date et de l'heure.

Le bus interne 101 est sorti vers l'extérieur du microcontrôleur 100, par l'intermédiaire du système de microfusibles 102. Selon une variante non illustrée, le système de microfusibles est évité en employant de la mémoire OTP EPROM interne. Cette variante permet les mêmes niveaux de sécurité que l'utilisation du système de microfusibles 102.

Le système de microfusibles 102 permet de réaliser un système de microcontrôleur programmable une seule fois. Les données nécessaires pour mettre en service les lecteurs LCL (clés de codages secrètes, numéros de série, identifiants, dates, heures), et le système d'exploitation du microcontrôleur 100 regroupant des programmes permettant aux lecteurs LCL de réaliser directement et/ou indirectement toutes les fonctionnalités relatives à la présente invention, sont programmés en usine dans la zone mémoire de mémoire non volatile (mémoire Flash 111). Le système d'exploitation est exécuté dans la mémoire DRAM 109.

La réalisation de la présente invention utilise de la mémoire FLASH comme support de stockage permanente pour le microcontrôleur 100. Le choix d'une telle mémoire FLASH peut permettre une mise à jour très facile du système d'exploitation initialement programmé en usine.

Selon la réalisation de la présente invention, les heures et les dates sont données, sauf mention contraire, par rapport au méridien origine des fuseaux horaires GMT (Greenwich Mean Time). Ainsi lors de la mise à l'heure durant la programmation du microcontrôleur 100 en usine, cette référence est prise pour l'horloge interne 104 du microcontrôleur 100 des lecteurs LCL.

Après programmation, le système de microfusibles est détruit ce qui empêche définitivement une nouvelle programmation du microcontrôleur 100, il n'y a plus alors d'accès directs à l'intérieur du microcontrôleur 100, car tous les contrôleurs et interfaces sont totalement sous le contrôle du processeur maître CPU1 (selon la construction du microcontrôleur 100). Le système d'exploitation du microcontrôleur 100 ainsi programmé est automatiquement chargé par le processeur maître CPU1 à chaque démarrage d'un lecteur LCL.

Ainsi l'intégralité des informations qui sont stockées dans la mémoire interne du microcontrôleur 100 sont sécurisées contre toutes tentatives de contrôle électronique externe au microcontrôleur 100. Compte tenu des propriétés physiques d'une pastille de silicium, de sa taille et de son boîtier, il offre dans l'état actuel de l'art une très bonne protection physique et logique contre toutes tentatives d'intrusions non autorisées dans les circuits internes du microcontrôleur 100. Selon des modes particuliers de réalisation, des techniques de protection supplémentaires peuvent toutefois être ajoutées autour du microcontrôleur 100. Une des techniques possibles concerne l'utilisation d'un dispositif électrique externe de protection de circuits intégrés, présenté

dans le brevet de la société IBM en 1990 (U.S. Patent 5,117,457).

Le système électronique esclave 120 sert à exécuter des programmes venant de l'extérieur du microcontrôleur, c'est à dire n'appartenant pas au système d'exploitation qui a été chargé dans la mémoire FLASH 111 lors de la programmation du microcontrôleur 100. Il permet d'exécuter des programmes à l'abri des attaques de virus informatiques éventuels. La sécurité est complète par rapport à ces attaques grâce à une protection physique caractérisée par une séparation logique d'une même pastille de silicium en deux parties 120 et 130 dont l'un 120 est esclave de l'autre 130.

Selon la réalisation de la présente invention, la zone mémoire DRAM 109 est strictement réservée à l'exécution du système d'exploitation du microcontrôleur 100 qui à été chargé en usine, lors de la procédure de programmation du microcontrôleur 100. Il sert aussi de mémoire tampon pour transférer les programmes et/ou données venant de l'extérieur du microcontrôleur vers la mémoire DRAM 110 via l'interface 106 contrôlée uniquement par le processeur maître CPU1. Les programmes venant de l'extérieur sont exécuter à partir de la mémoire DRAM 110, par le processeur esclave CPU2 contrôlé par le processeur maître CPU1 via le contrôleur 113.

Selon la réalisation de la présente invention, par rapport l'utilisations des lecteurs LCL avec différents types d'ordinateur et de systèmes d'exploitations, et compte tenu de l'écriture des fonctions F_0 précédemment décrites, le CPU2 est un processeur Java (PicoJava) de la société Sun Microsystems. Cette caractéristique rend le développement des fonctions $\{F_0, F_1,\ldots,F_i, \ldots F_n\}$ indépendant des ressources de calculs de l'ordinateur qui exécute un logiciel protégé selon la présente invention, et des ressources internes du LCL. De plus, le concepteur de logiciels protégés par la présente invention, peut tester le fonctionnement des fonctions $\{F_0, F_1,\ldots,F_i, \ldots F_n\}$ indépendamment du lecteur LCL avec un programme simulant une machine virtuelle JAVA.

Selon la réalisation de la présente invention, CPU1 est un processeur du type 80386SX. Il peut donc utiliser directement via son bus interne 101 une grande quantité de mémoires internes et/ou externes. Sa capacité de calculs permet d'envisager une forte capacité de traitements d'informations en multitâche.

Selon des modes particuliers de réalisation, le microcontrôleur peut ne pas utiliser de processeurs JAVA, mais un processeur du type 80386SX dont le système d'exploitation serait la machine virtuelle JAVA de la société Sun Microsystems que le CPU1 chargerait à chaque nouvelle exécution de programmes dans la DRAM 110 afin d'éviter d'éventuelles attaques de virus informatiques.

De plus, selon la réalisation de la présente invention, la totalité des informations éventuelles de la mémoire DRAM 110 est effacée par CPU1 avant chaque nouveau chargement de programmes qui doivent être exécutés par CPU2. CPU1 met en pause CPU2 par l'intermédiaire du contrôleur de CPU2 113, et efface, par exemple par une désactivation momentanée des circuits qui composent le module de mémoires DRAM 110, le contenu de cette mémoire DRAM 110 grâce à l'interface 106. Ensuite, CPU1 charge les paramètres d'exécution et le nouveau programme à exécuter dans la DRAM 110 directement grâce au contrôleur DMA 107. Ce chargement direct permet de ne pas

utiliser CPU2 et de permettre au CPU1 de contrôler complètement la DRAM 110. Ainsi, après chargement du programme, CPU1 donne alors la main à CPU2 par l'intermédiaire du contrôleur de CPU2 113. CPU2 exécute alors le nouveau programme. Ainsi compte tenu de toutes ces mesures, si ce programme est un virus informatique volontairement inscrit dans une des fonctions de type F_i, il ne pourra toutefois porter aucunes atteintes au fonctionnement du microcontrôleur 100, ni recopier vers l'extérieur des données non effacées concernants les anciennes fonctions qui ont été exécutées par CPU2 dans la mémoire DRAM 110. De plus, CPU1 conserve le contrôle des accès aux données contenues sa partie 130. Cette procédure de chargement de programme et/ou de données dans la DRAM 110, est répétée pour chaque programme qui doit être exécuté par le processeur esclave CPU2.

Selon la réalisation de la présente invention, l'architecture présentée par la figure 3, permet d'empêcher un programme n'appartenant pas au système d'exploitation du microcontrôleur 100 d'effectuer des lectures et/ou modification dans la mémoire interne du système 130 intégré dans le microcontrôleur 100. Il permet aussi d'empêcher ces programmes de contrôler les interfaces et/ou contrôleurs du microcontrôleur 100, et donc d'empêcher des pirates de lire le contenu des mémoires sécurisées physiquement et logiquement du système 130, intégrées dans le microcontrôleur 100.

De plus, selon la présente invention, l'interface bus externe 105 permet au microcontrôleur 100 de contrôler des périphériques externes reliés au bus externe 114. Ce bus permet d'ajouter au LCL des périphériques d'enregistrement comme par exemple un média d'enregistrement du type Flash Disk (mémoire FLASH utilisé un disque standard), un périphérique de communication réseau.

L'architecture du microcontrôleur 100 permet une grande modularité de fonctionnement. Selon la réalisation de la présente invention, on connecte sur ce bus 114 un périphérique d'accès réseau Ethernet pour une communication en protocole TPC/IP. Selon le type de communication utilisé entre un lecteur LCL et un ordinateur donné, on pourra connecter sur ce bus un périphérique adéquat à cette communication. Ainsi, on peut ajouter un périphérique radio récepteur 22, pour utiliser un lecteur LCL dans le contexte 20 de la figure 1. L'usage de ce récepteur sera défini par la suite.

Selon la réalisation de la présente invention, compte tenu du contrôleur PCMCIA 154 intégré dans le microcontrôleur 100, les périphériques utilisés peuvent être aussi des cartes PCMCIA utilisée par le microcontrôleur 100 pour toutes opérations relatives à la présente invention. Ces cartes PCMCIA peuvent être des cartes Ethernet PCMCIA, des cartes FLASH PCMCIA, un disque dur PCMCIA, une carte module de réceptions numériques hertziennes PCMCIA. Ces différentes cartes ne sont pas illustrées. L'usage du contrôleur PCMCIA permet d'ajouter à un lecteur LCL donné des périphériques plus facilement par rapport au bus externe 114. Le port USB 150 sert à une connexion à grande vitesse de transmission et de réceptions entre un ordinateur, et un lecteur LCL donné. Il s'agit des contextes 30 et 20.

Le port d'E/S 151 permet selon la réalisation de la présente invention, de communiquer avec une carte CL.

En référence à la figure 5, la carte électronique CL 60 est construite autour d'un microcontrôleur 400 intégrant sur une même pastille de silicium un processeur CPU 405 relié par un bus interne 406 à un module de mémoires Flash 401, un module de mémoires OTP EPROM 407, un module de mémoires DRAM 404, un Port Série E/S RS232 403, un contrôleur de CompactFlash 402 de la société SanDisk.

Selon une variante non illustrée, le microcontrôleur intègre sur une même surface de pastille de silicium un coprocesseur de cryptage DES pour permettre au CPU d'effectuer des opérations de cryptage plus rapidement.

Selon la réalisation de la présente invention, les accès en lecture dans le module de mémoires OTPEPROM directement de l'extérieur du microcontrôleur 400 sont supprimés, afin de laisser tous les accès aux circuits internes du microcontrôleur 400 sous le contrôle unique du processeur CPU 405. La présente invention ne requiert pas l'utilisation d'une grande puissance de calcul au niveau du processeur CPU 405 intégré dans le microcontrôleur 400.

Ce microcontrôleur 400 comporte un moyen de sécuriser la lecture et/ou la modification illicite des informations qui sont contenues dans sa mémoire interne. Une méthode de sécurisation des accès en mémoires est proposée dans le brevet U.S. Pat. 5,293,424 daté du 8 mars 1994.

Selon la présente invention, la mémoire OTP EPROM interne 407 sert à stocker des clés de cryptage, des numéros de série d'identification, des dates, le système d'exploitation du microcontrôleur 400 réalisant directement et/ou indirectement des fonctions relatives à son utilisation selon la présente invention.

Selon la réalisation de la présente invention, la mémoire FLASH interne 401 sert à stocker de manière permanente des données supplémentaires après la sortie d'usine de la carte CL. Elle sert aussi à stocker des programmes supplémentaires qui permettent au microcontrôleur 400 de réaliser directement et/ou indirectement des fonctions supplémentaires relatives à son utilisation selon la présente invention après la sortie d'usine de la carte CL.

Selon la réalisation de la présente invention, et de manière générale, la carte CL est alimentée électriquement par LCL lorsqu'elle est connectée à LCL. Cette liaison électrique n'est pas illustrée.

Selon la présente invention, le microcontrôleur 400 est protégé contre toute modification des informations qu'il contient. Il s'agit d'un microcontrôleur similaire à ceux des cartes à puces. Il est relié à un connecteur femelle 63 qui permet de communiquer par contact avec un lecteur LCL donné.

Selon la présente invention, la carte CL est un appareil portatif de petite taille possédant une unité de stockage amovible de forte capacité.

Selon la réalisation de la présente invention, le microcontrôleur 400 est relié à la sortie de son contrôleur de CompactFlash à un support de connexion 64 pour modules de mémoires de type CompactFlash.

Selon des modes particuliers de réalisation, le contrôleur de CompactFlash peut ne pas être intégré sur la même pastille de silicium que le microcontrôleur 400. Il peut aussi utiliser un autre média d'enregistrement telle que des modules DiskOnChip de la société M-Systems ou tout autre système propriétaire et amovible de mémoires non volatiles.

Dans l'état actuel de l'art, des microcontrôleurs possédant les caractéristiques du microcontrôleur 400 sont très nombreux. Selon la réalisation de la présente invention, un microcontrôleur 32 bits RISC présentant les caractéristiques du microcontrôleur 400 à été intégré sur une même pastille de silicium avec le contrôleur de carte CompactFlash. Cette intégration utilise les technologies d'intégrations ASIC (Application Specific Integrated Circuit).

Ainsi, l'intégralité des informations qui sont stockées dans la mémoire interne du microcontrôleur 400 est sécurisée contre toutes tentatives de contrôles électroniques externes.

Selon la présente invention, la carte CL est un appareil portatif de petite taille. Il permet de transporter les informations concernant l'utilisation de logiciels protégés, indépendamment du lecteur électronique LCL. Elle est utilisée comme une clé d'accès à l'utilisation de logiciels protégés selon la présente invention. Sa portabilité permet à un utilisateur d'utiliser les logiciels dont il a acheté les droits d'utilisation (licences) sur tout ordinateur qui posséderait ces logiciels.

Selon la réalisation de la présente invention, le format géométrique de CL est compris entre celui de la carte CompactFlash est celui d'une carte PCMCIA.

En référence à la figure 6, un trou 62 a été placé dans un coin du boîtier 60 représentant CL. Ainsi, la carte CL peut être attachée à un porte-clés mécanique. Selon la réalisation de la présente invention, le numéro de série ID.c non illustré d'une carte CL donnée est imprimé sur le boîtier 60 de CL.

De plus, selon la présente invention, en référence à la figure 7, le module de mémoires CompactFlash 61 est détachable du boîtier 60 via le système de support de connexion 64 pour carte CompactFlash.

En référence à la figure 8 et selon la réalisation de la présente invention, la carte CL se connecte par contact avec un lecteur LCL via un jeu de deux connecteurs mâle femelle. Une carte CL possède un connecteur femelle 63 qu'il connecte sur le connecteur 210 mâle correspondant du lecteur LCL. Ces jeux de connecteurs permettent une communication série RS232 entre LCL et CL par contact. De plus, il permet d'apporter de l'énergie électrique aux circuits électroniques de la carte CL. Selon la réalisation de la présente invention, le lecteur LCL est alimenté sur secteur.

Selon des modes particuliers de réalisation non illustrés, CL peut avoir sa propre alimentation électrique pour permettre un fonctionnement autonome. Selon ces modes particuliers de réalisation, on peut intégrer dans CL un module de communication hertzienne ou infrarouge pour permettre des communications sans contacts avec LCL. Un exemple d'un tel dispositif est apporté par Hough dans U.S. Pat. No. 5,412,253. Bien sûre, le lecteur LCL possède dans ces conditions les ports de communications adéquats.

Selon la réalisation de la présente invention, toutes les clés de codages écrites en usine lors de

la programmation des appareils de type LCL et CL sont des clés de 128 bits définies par rapport à l'algorithme DES. Ainsi, compte tenu de la présente description de l'invention, un module de mémoires OTP EPROM de 256 kilooctets, un module de mémoires Flash de 64 kilooctets, un module de mémoire DRAM de 512 kilooctets ont été intégré avec le CPU 405 (processeur RISC), le Port Série E/S RS232 403 et le contrôleur de carte CompactFlash sur une même pastille de silicium. Bien entendu, d'autres tailles plus grandes de mémoires peuvent être utilisées en fonction des disponibilités des macros d'intégration ASIC et de leur coût. Ces quantités sont données par rapport à la présente réalisation.

Par ailleurs, selon la réalisation de la présente invention, la taille retenue pour le module de mémoire Flash 111, est de 1 mégaoctets. La taille retenue pour le module de mémoire DRAM 109, est de 2 mégaoctets. La taille retenue pour le module de mémoire DRAM 110 est de 1 mégaoctet. Ces quantités sont données par rapport à la présente réalisation.

L'ensemble des programmes relatifs aux fonctionnalités du lecteur LCL constitue le système d'exploitation interne du microcontrôleur 100. Ce système d'exploitation est enregistré dans la mémoire Flash 111 du microcontrôleur 100 lors de sa programmation en usine.

L'ensemble des programmes relatifs aux fonctionnalités de la carte CL constitue le système d'exploitation interne du microcontrôleur 400. Ce système d'exploitation est enregistré dans la mémoire OTP EPROM 407 du microcontrôleur 400 lors de sa programmation en usine.

Selon la présente invention, les communications entre LCL et CL sont sécurisées. La présente invention concerne l'utilisation d'une méthode d'authentification permettant à un groupe d'appareils quelconques de se reconnaître à l'aide de cette méthode. Cette authentification permet que seuls les appareils relatifs à la présente invention, certifiés par l'organisme qui gère le serveur aSVR, puissent fonctionner ensemble. Cette méthode selon la présente invention, permet d'empêcher que des appareils non reconnus par l'organisme qui gère les appareils relatifs à la présente invention, ne puissent fonctionner avec ceux reconnus. Cette méthode empêche ainsi des appareils pirates d'effectuer des lectures de données dans les mémoires électroniques sécurisées des appareils relatifs à la présente invention.

La présente invention concerne des appareils qui ne peuvent être utilisés que pendant une durée déterminer dans le temps. Pour la réalisation, les appareils LCL et CL sont tous caractérisés par une date DB de mise en service et une date DE qui indique que l'utilisation de ces appareils relatifs à la présente invention, expire fin DE. DB et DE constitue une information publique non modifiable.

Ainsi, selon la réalisation de la présente invention, la date DB du lecteur LCL, notée DB.d, est écrite lors de la programmation en usine du microcontrôleur 100, dans une zone libre de la mémoire Flash 111. De plus, la date DE d'un lecteur LCL, notée DE.d, est écrite lors de la programmation en usine du microcontrôleur 100, dans une zone libre de la mémoire Flash 111.

De plus, selon la réalisation de la présente invention, la date DB de CL, notée DB.c, est écrite lors de la programmation en usine du microcontrôleur 400, dans une zone libre de la mémoire

Flash 401. De plus, la date DE d'une carte CL, notée DE.c, est écrite lors de la programmation en usine du microcontrôleur 400, dans une zone libre de la mémoire OTPEPROM 407.

Ainsi, selon la réalisation de la présente invention, l'organisme qui gère aSVR génère pour chaque semaine du calendrier international qui commence le lundi et qui se termine fin dimanche une clé kLi. La clé kL1 est la première clé qui a été générée pour la première semaine où les premiers appareils relatifs à la présente invention, ont été mise en service. La clé kLi indique la clé de la semaine i par rapport à cette première semaine. Toutes ces clés sont entièrement créées et gardées secrètement par l'organisme qui gère le serveur aSVR afin de garantir la sécurité d'utilisation des appareils relatifs à la présente invention.

Selon la réalisation de la présente invention, pour une carte CL considérée, lors de la procédure de programmation de son microcontrôleur 400, la clé secrète de codage kLj est écrite dans une zone libre de la mémoire OTPEPROM 407. Cette clé secrète kLj correspond à la semaine j par rapport à la première semaine où les premiers appareils relatifs à la présente invention, ont été mise en service. La clé kLj a été choisie de telle sorte que la semaine j contient la date DB de la mise en service de cette carte CL. Cette clé secrète ne sera jamais révélée à l'utilisateur. Par ailleurs, elle est connue uniquement par l'organisme qui gère le serveur aSVR.

De plus, selon la réalisation de la présente invention, pour un lecteur LCL considéré, lors de la procédure de programmation de son microcontrôleur 100, la liste de clés secrètes de codage $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$ correspondant à toutes les clés associées aux semaines comprises entre la date DB.d diminuée de 1460 jours et la date DE.d, sont stockées dans la mémoire Flash 111 du microcontrôleur 100. Bien entendu, leurs adresses mémoires suivent une convention adoptée pour permettre de les retrouver dans la mémoire 111. Ces clés secrètes sont par ailleurs connues uniquement par l'organisme qui gère le serveur aSVR. Toutes ces clés secrètes qui viennent d'être citées, sont générées par rapport à l'algorithme de cryptage DES. Chacune de ces clés secrètes a une taille de 128 bits.

Selon la réalisation de la présente invention, la durée qui sépare une date de mise en service DB et une date de fin d'utilisation DE des appareils relatifs à la présente invention, est de 1461 jours (4 ans). Ainsi $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$ n'occupe pas plus de 7 000 octets (approximation volontairement excessive) dans la mémoire Flash du microcontrôleur 100 d'un LCL donnée. La figure 4 permet de comprendre le choix du nombre de clés dans la liste $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$. Ce nombre est du à l'existence des appareils les plus anciens, encore en service à la date DB.d de mise en service du LCL considéré. Compte tenu de la capacité des mémoires Flash intégrées au sein d'un microcontrôleur 100, la totalité des clés $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$ peut donc être stockée avec le système d'exploitation microcontrôleur 100.

Ainsi, la réalisation de ladite procédure d'authentification est basée sur l'utilisation judicieuse de toutes ces clés. Le nombre de clés employées permet à ce que si une clé venait à être cassée, le fonctionnement lié à l'authentification par rapport à cette clé, ne mette pas en échec l'ensemble du système relatif à la présente invention.

En référence à la figure 10, pour l'utilisation de LCL, dans un premier temps, un lecteur LCL ne peut fonctionner que si la date courante indiquée par l'horloge temps réel interne 104 du microcontrôleur 100 de LCL est comprise entre les dates DB.d et DE.d du même LCL. Autrement la suite ne peut aboutir 551. Cette condition est illustrée sur la figure 10 par l'élément 501

Dans un deuxième temps, en référence à l'étape 502, pour mettre en fonctionnement un lecteur LCL, l'utilisateur doit effectuer une opération d'identification détaillée par la suite.

Dans un troisième temps, pour mettre en fonctionnement une carte CL, l'utilisateur doit connecter d'abord sa carte CL (étape 503) sur le support de type mâle de connexion 210 possédé par le lecteur LCL, pour permettre une communication par contact entre le lecteur LCL et la carte CL. Le support de type mâle 210 du lecteur LCL est relié au port E/S RS232 151 du microcontrôleur 100. La carte CL possède donc un connecteur femelle 63 relié au port E/S RS232 403 de son microcontrôleur 400. L'utilisateur doit effectuer ensuite une opération d'identification décrite par la suite.

Selon la réalisation de la présente invention, dans un quatrième temps 504, le microcontrôleur 400 de la carte CL envoie sous une forme non codée sa date de mise en service DB.c au microcontrôleur du LCL via ladite liaison RS232. Si un pirate modifier la valeur DB.c transmise, la suite de la présente description ne pourra aboutir avec succès.

De l'autre côté et après réception de DB.c, dans un cinquième temps 505, le processeur CPU1 du microcontrôleur 100 de LCL effectue une correspondance entre DB.c et une clé secrète notée kLj.d éléments de la liste $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$ de telle sorte que la semaine j associée à kLj.d selon la présente invention contienne DB.c.

Dans un sixième temps 506, le processeur CPU1 génère une clé kCS de 128 bits par rapport au cryptage DES, à l'aide de son générateur de nombres aléatoires 112. kCS est conservées secrètement dans la mémoire interne DRAM 109 du microcontrôleur 100.

Selon la présente invention, kCS est codée ensuite par kLj.d puis envoyée sous sa forme codée, notée ekCS, vers le microcontrôleur de CL.

Dans un septième temps 507, à réception de ekCS, la carte CL tente de décoder ekCS avec sa clé secrète kLj. Selon la présente invention, si le décodage réussit, ladite procédure d'authentification a réussi. Le microcontrôleur de CL utilisera alors la clé kCS pour envoyée des informations codées vers LCL, et pour décodée des informations venant par la suite de LCL. Le microcontrôleur 400 de la carte CL stocke dans sa mémoire interne sécurisée DRAM 404 cette clé secrète kCS.

Ainsi, selon la réalisation de la présente invention, dans un huitième temps 508, le microcontrôleur de CL envoie ensuite la date d'expiration DE.c associée à CL sous une forme codée par la clé kCS vers le microcontrôleur du lecteur LCL.

A réception, dans un neuvième temps 509, CPU1 vérifie si DE.c n'est pas dépassé par rapport à la date courante donnée par l'horloge interne du microcontrôleur 100 de LCL. Si cette date est dépassée, LCL refusera de poursuivre la communication avec la carte CL 552. Sinon, une

communication sécurisée entre LCL et CL peut avoir lieu 559.

Ainsi, chaque session de communication entre LCL et CL qui commence par leur connexion par contact et qui se termine lorsque l'une des conditions suivantes est remplie : CL est déconnecté de LCL, la date courante indiquée par l'horloge 104 a dépassé la date DE.c, la date courante indiquée par l'horloge 104 a dépassé la date DE.d. La déconnexion de la carte CL du lecteur LCL est marquée par l'absence de charge à la sortie de l'alimentation électrique utilisée pour alimenter les circuits électroniques de la carte CL.

De plus, selon la présente invention, toutes les communications entre le lecteur LCL et la carte CL sont sécurisées par l'utilisation d'une méthode de cryptage symétrique utilisant une clé secrète, notée kCS.

Selon la réalisation de la présente invention, l'utilisateur qui désire faire fonctionner un appareil relatif à la présente invention, doit saisir sur le clavier de son lecteur LCL contrôlé par le microcontrôleur 100 par l'intermédiaire du contrôleur de claviers et écrans LCD 155, un code PIN. Au moment de sa saisie, les chiffres tapés seront inscrits sur l'écran LCD non illustré et contrôlé par le contrôleur 155. Ce code lui a été fourni lors de la première acquisition (l'achat) de l'appareil en question. Le code PIN est un code numérique de 5 chiffres associé à chaque appareil. Il doit être conservé secrètement par le propriétaire de l'appareil correspondant. L'utilisation d'un tel code est similaire à celui utilisé avec des méthodes d'identification présente dans le monde des cartes à puces (SmartCard) dans l'état actuel de l'art. Les étapes qui permettent de vérifier la saisie correcte du code PIN associé à chaque appareil selon la présente invention, sont évidente et ne sont pas détaillée dans la présente description de l'invention. Il faut toutefois ajouter que le code PIN d'une carte CL est saisi sur le clavier du LCL, puis transmis sous sa forme non codée, vers le microcontrôleur 400 de la carte CL, via la liaison série RS232 présente entre une carte CL donnée et un lecteur LCL donnée. Il est donc sous-entendu, sauf mention contraire, dans un fonctionnent correct d'un appareil relatif à la présente invention, que la saisie du code PIN a été menée avec succès.

La présente invention concerne l'utilisation d'une carte CL comme média d'enregistrement portatif, sécurisé contenant des fichiers d'autorisation d'utilisation de tous les logiciels protégés selon la présente invention, et acquis légalement par l'utilisateur et séparément de l'acquisition du logiciel (le média d'enregistrement). Ces fichiers ont été enregistrés dans la carte CL en suivant une procédure d'acquisition de logiciels décrite par la suite.

Selon la réalisation de la présente invention, un fichier d'autorisations d'utilisation d'un logiciel donné ayant le numéro de série S#, noté Fich.S#, est défini comme un fichier binaire dont les bits de données sont ordonnés de la façon suivante : S# du logiciel (128 bits), ID.c (128 bits), nombre de licences (L#.S# : 16 bits), dernière utilisation (DR.S# : Jour 5 bits, mois 4 bits, année 12 bits, heure 5 bits, minutes 6 bits, secondes 6 bits), première utilisation (DP.S# : date et heure 38 bits), durée d'utilisation courant (DU.S# en minutes : 24 bits), nombre d'exécutions du logiciel (28 bits), données diverses (Misc :1024 bits), clé kEL.S (128 bits), clé kX.S# (128 bits). Le total est de

1680 bits, soit un fichier de 210 octets.

Selon la réalisation de la présente invention, une carte de type CompactFlash 61 distribuée par la société SanDisk d'une capacité de stockage de 4 mégaoctets est insérée comme indiqué sur la figure 7 sur un support de connexion 64 adapté et présent sur la carte CL, pour permettre la connexion de cette carte CompactFlash vers le contrôleur de cartes CompactFlash 402 du microcontrôleur 400. Les cartes CompactFlash sont compatibles avec le standard ATA des cartes PCMCIA dans l'état actuel de l'art. Ces modules de mémoires CompactFlash sont utilisés comme des disques de stockage d'informations. Les instructions nécessaires concernant l'écriture du driver qui permet au microcontrôleur 400 d'utiliser le module CompactFlash de 4 mégaoctets sont donnée par le standard ATA. Ce driver non illustré, permet selon la réalisation de la présente invention, au microcontrôleur d'effectuer les opérations suivantes sur la carte CompactFlash : lectures de fichiers, modifications de fichiers, créations de fichiers.

Ainsi, selon la réalisation de la présente invention, la carte CL permet de stocker plus de 10000 fichiers d'autorisations d'utilisation de logiciels protégés selon la présente invention. Ce nombre est largement suffisant pour tous les logiciels protégés par la présente invention, qu'un utilisateur peut acquérir légalement. Toutefois, l'utilisateur pourra changer de cartes CompactFlash pour une plus grande capacité de stockage. Compte tenu de la norme ATA, ce changement ne nécessite ni de mise à jour des programmes système du microcontrôleur 400, ni de changement du boîtier 60 et du support pour carte CompactFlash 64.

Selon la présente invention, les informations stockées sur la carte CL, concernant les autorisations d'utilisation de logiciels protégés selon la présente invention, ne dépendent pas du média d'enregistrement mais de l'entité carte électronique CL. Ainsi, un utilisateur d'une carte CL pourra utiliser plusieurs cartes CompactFlash pour stocker des fichiers de type Fich.S#. Les données qui sont stockées sur une première carte CompactFlash associée à une carte CL donnée peuvent être transférées vers une deuxième carte CompactFlash. Cette opération est effectuée à l'aide du programme PGM précédemment cité.

Selon la présente invention, les fichiers d'autorisations Fich.S# sont stockés sous une forme codée, notées eFich.S#, avec une clé secrète kS.c de 128 bits par rapport à la technique de cryptage DES, dans la carte CompactFlash utilisée comme un disque de stockage. La clé secrète kS.c a été inscrite dans la mémoire OTPEPROM 407 lors de la programmation en usine du microcontrôleur 400.

Ainsi, eFich.S# ne pourra être utilisé que par la carte CL qui l'a créé, car la clé secrète kS.c est différente pour chaque carte CL mise en service.

De plus, la présente invention concerne une méthode qui permet de séparer l'acquisition du média d'enregistrement contenant un ou des logiciel(s) protégé(s) par la présente invention, du droit d'utilisation de ce ou ces logiciel(s).

Selon la réalisation de la présente invention, les médias d'enregistrement de logiciels protégés selon la présente invention, sont librement distribués. Cependant, un logiciel protégé selon la

présente invention, ne peut être exécuté uniquement qu'après une opération d'acquisition légale d'un fichier Fich.S# d'autorisation d'utilisation de ce logiciel numéroté S#. Ainsi, en référence à la figure 2, lorsqu'un utilisateur désire obtenir une ou des licences d'utilisation, il doit connecter tout d'abord son lecteur LCL avec le serveur aSVR, à l'aide du programme PGM. Selon la réalisation de la présente invention, cette connexion est établie via le réseau Internet.

Pour commencer à acquérir une autorisation (une ou des licence(s)) d'utilisations d'un logiciel numéroté S# et protégés selon la présente invention, l'utilisateur commence par connecter sa carte CL sur ledit lecteur LCL donné. L'utilisateur doit maintenir sa carte CL connectée au moins jusqu'à la fin de la présente procédure d'achat en ligne (avec connexion). On suppose que l'utilisateur veut acquérir NL nombre(s) de licences d'utilisation de ce logiciel numéroté S#. Il saisit ensuite le code PIN de sa carte CL sur le clavier dudit lecteur LCL. L'opération d'authentification précédemment citée doit aboutir avec succès pour continuer.

Ainsi, en référence à la figure 11, selon la réalisation de la présente invention, au début de la connexion avec aSVR, ledit lecteur LCL communique à aSVR son numéro ID.d (étape 601). Dans ce contexte de communication, aSVR est le système distant représenté sur la figure 2. Bien sûre, les opérations commerciales qui sous-entend que aSVR accepte une communication avec ledit lecteur LCL sont sous-entendues. Le programme PGM envoie ensuite vers aSVR le numéro de série S# dudit logiciel, saisie par l'utilisateur (acheteur), et le nombre NL. Ces deux informations sont envoyées sous une forme codée par la clé secrète kT.d du lecteur LCL de l'utilisateur.

De plus, la réalisation de la présente invention considère que le concepteur dudit logiciel au numéro de série S# dispose lui aussi d'un serveur noté dSVR, non illustré et relié par le réseau Internet à aSVR. Le serveur dSVR est connecté de son côté avec le lecteur LCL dudit concepteur. Bien entendu, cela suppose que le système d'exploitation d'un lecteur LCL soit programmé de telle sorte qu'il puisse répondre à certaine requête relative à la présente procédure d'achat du serveur dSVR. Ainsi (étape 602), dSVR communique ensuite à aSVR le numéro ID.d de son lecteur LCL qui a servi à créer le logiciel S# protégé selon la présente invention. La communication entre dSVR et le lecteur LCL se passe comme sur la figure 2 où l'ordinateur 50 est représenté ici par le serveur dSVR.

Pour permettre une communication sur deux niveaux, le lecteur LCL de l'utilisateur communique à aSVR (étape 603) sous une forme codée une clé publique, notée kP. La clé kP est codée avec la clé secrète kT.d du LCL de l'utilisateur. kP est une clé publique, relative à la technique de cryptage RSA (Rivest, Shamir et Adleman). La clé kP est créée avec sa clé privée kV pour l'occasion (dynamiquement) et effacée à la fin de cette procédure d'acquisition de licences. La forme codée par ladite clé kT.d de kP est notée ekP.kT. De plus, il faut noter que aSVR ne connaît pas la valeur de la clé privée kV associée à kP. Le codage systématique des informations lors de la communication évite d'éventuelles modifications des données échangées lors de leur transfert.

A réception, aSVR décode ekP.kT à l'aide des informations qu'il possède concernant le

lecteur LCL de l'utilisateur. Selon la réalisation de la présente invention, aSVR est seul en dehors de ce lecteur LCL, à connaître la correspondance du numéro ID.d avec la clé kT.d par rapport à un lecteur LCL donné. La clé kP est ensuite codée avec la clé secrète kT.d relative au lecteur LCL du concepteur du logiciel S# protégé selon la présente invention. La nouvelle forme codée de kP est notée ekP.kT2. Ensuite, aSVR communique ekP.kT2 604 au lecteur LCL du concepteur via le serveur dSVR. De cette manière, l'utilisation des techniques de protection de logiciels selon la présente invention, crée une dépendance entre le concepteur de logiciels protégés selon la présente invention, et l'organisme qui gère aSVR. Ainsi, Le concepteur de logiciel n'a pas à constituer des stocks autre qu'un stock de médias d'enregistrement contenant le logiciel protégé selon la présente invention.

A réception, le lecteur LCL dudit concepteur décode ekP.kT2 avec sa clé kT.d. Avec la clé public kP, LCL relatif au concepteur code ensuite kEL.S# généré lors de la procédure de création du logiciel S# protégé selon la présente invention avec la clé kP. La forme codée de kEL.S# par kP est notée ekEL. Selon des variantes non illustrées, après avoir décodé ekP.kT2, ledit peut communiquer kP au serveur dSVR afin de lui laisser le codage de la clé kEL.S# par la clé kP. Mais ces variantes ne changent en rien quant au principe fondamentale de la présente invention.

Selon la réalisation de la présente invention, dSVR reçoit ensuite de aSVR la valeur de NL. NL permet au concepteur de logiciels protégés selon la présente invention, d'effectuer une comptabilité avec l'organisme qui gère aSVR. Selon une variante de la présente procédure d'acquisition d'autorisation d'utilisations, dSVR envoie après réception de la clé kP, une clé public kPUB relatif au cryptage RSA vers le LCL de l'utilisateur (acheteur) via aSVR. Cette variante permet au programme PGM connecté avec le lecteur LCL de l'utilisateur, de retourner à dSVR par l'intermédiaire de aSVR, la valeur codée avec kPUB de NL. Cette variante permet à ce que dSVR ne se fie pas à la bonne sincérité de aSVR. Ainsi, cette variante permet à dSVR de contrôler exactement le nombre de licences vendues.

Selon la réalisation de la présente invention, dSVR envoie ensuite (étape 605) ekEL à aSVR. Ce serveur aSVR envoie ensuite (étape 606) kX.S# sous une forme codée avec la clé kT.d (celui du lecteur LCL de l'utilisateur) vers le lecteur LCL de l'utilisateur. La clé kX.S# est une clé créée et stockée par aSVR lors de la procédure de protection précédemment décrite du logiciel LD numéroté S#. Le serveur aSVR envoie ensuite vers le lecteur LCL de l'utilisateur, eKEL. A réception, le lecteur LCL de l'utilisateur décode eKEL par kV. Il obtient donc kEL.S#. Ce lecteur LCL décode aussi la forme codée de kX.S# par sa clé kT.d.

Selon la réalisation de la présente invention, ledit lecteur LCL envoie ensuite à ladite carte CL, sous une forme codée avec la clé kCS obtenue lors de la procédure d'authentification précédemment décrite, les données suivantes : S#, NL qui correspond au nombre de licences demandées par l'utilisateur à aSVR, kX.S#, kEL.S#, la date et l'heure courantes indiquées par l'horloge 104.

Ainsi, le microcontrôleur 400 de la carte CL décode les différentes données reçues avec kCS.

Le microcontrôleur 400 procède alors à la procédure (étape 607) de mise à jour qui suit. Le microcontrôleur 400 vérifie si un fichier codé eFich.S# d'autorisation d'utilisation existe déjà pour le logiciel en question numéroté S#. Dans ce cas, il procède à sa modification en augmentant la valeur du champ L#.S# dans le fichier Fich.S# correspondant, de la valeur de NL. Dans le cas où l'utilisateur aurait acquis une autorisation dépendante du temps, bien entendu des mises à jour sur les champs correspondants sont effectuées dans le fichier Fich.S#. Pour la compréhension de la présente invention, certain point évident sont sous-entendu.

S'il n'existe pas de fichier Fich.S# correspondant, un nouveau fichier Fich.S# est créé. Ainsi, selon la réalisation de la présente invention, le microcontrôleur 400 de la carte CL crée le fichier Fich.S# en remplissant les champs suivants du nouveau fichier : S#, L#.S#, ID.c numéro de série de la carte CL en question, kEL.S#, kX.S#. Les valeurs de DR.S# et de DP.S# sont initialisées par la valeur de l'heure et de la date courante. Les champs DU.S# et « nombre d'exécutions du logiciel » prennent évidemment la valeur nulle. Le champ Misc sert à stocker des valeurs liées à des besoins éventuels de définir des champs d'informations supplémentaires. Misc est initialement à zéro. L#.S# prend la valeur de NL.

Selon des modes particuliers de réalisation non décrites, un utilisateur peut directement se connecter sur le serveur dSVR pour acheter des licences lorsque aSVR a communiqué la valeur de kX.S# d'un logiciel numéroté S# protégé selon la présente invention. Cette variante permet de décentraliser la vente des licences.

Compte tenu du format des fichiers Fich.S# et de sa forme codée eFich.S#, seul une carte CL donnée et numérotée ID.c peut utiliser les fichiers Fich.S# correspondant à ID.c

Selon la réalisation de la présente invention, on considère un logiciel, noté LD, ayant subi une procédure de protection selon la présente invention. Bien entendu, les explications qui suivent sont valables pour tout logiciel protégé selon la présente invention. Pour permettre une explication, on considère son utilisation dans le cas où l'ordinateur hôte est connecté à un lecteur LCL via un réseau de type Ethernet, en TCP/IP. C'est le cas du contexte 40. Le driver DRV précédemment citée permet de communiquer avec LCL par rapport au protocole TCP/IP. Selon la réalisation de la présente invention, la couche TCP/IP est sous-entendu dans les propositions de communication entre le programme driver DRV et le lecteur LCL considéré.

Selon des modes particuliers de réalisation non décrite, mais illustrée par la figure 4, plusieurs LCL peuvent être présents sur le réseau. Cependant, ces organisations ne change pas les caractéristiques de la présente invention.

Selon la réalisation de la présente invention, à l'obtention d'une ou plusieurs autorisation(s) d'utilisation du logiciel LD (licences d'utilisation), LD peut alors être exécuté.

Ainsi, au lancement du logiciel LD sur un ordinateur hôte, la fonction FF_0 est exécutée en premier. De manière générale, selon la réalisation de la présente invention, toutes les fonctions FF_i effectuent toutes une procédure commune : le chargement de eF_i vers LCL.

Selon la réalisation de la présente invention, lorsque le logiciel LD appelle une fonction donnée FF_i, il lui transmet par passage de paramètres en utilisant par exemple la pile du système de l'ordinateur hôte, des informations paramètres, notées PARAM, utilisées directement et/ou indirectement lors de l'exécution éventuelle de la fonction F_i correspondant à FF_i. Ensuite, FF_i charge en mémoire le contenu du fichier eF_i décrit précédemment. Puis, FF_i appelle (au sens d'un appel de programmes informatiques) le driver DRV afin de lui communiquer les informations PARAM et l'adresse mémoire où se trouve eF_i. Ces informations sont ensuite envoyées vers LCL à travers le réseau considéré. Des paramètres supplémentaires peuvent être rajoutés par une fonction FF_i dans les paramètres PARAM. Ainsi, PARAM contient notamment des informations relatives à l'heure courante dans l'ordinateur qui exécute ladite fonction FF_i. Il contient aussi un identifiant unique représentant cet ordinateur hôte (par exemple l'adresse IP de cet ordinateur sur le réseau qui est fournie par le système d'exploitation de cet ordinateur) et le numéro de série S# du logiciel correspondant à ladite fonction FF_i. Bien entendu, dans le cas d'un lecteur directement connecté sur un ordinateur (c'est le contexte 30 et 20), il n'y a pas de problèmes d'identifiants uniques. Ainsi, on note l'identifiant unique de l'ordinateur IDIP. Cette variable est utilisé par le microcontrôleur 100 du lecteur LCL pour gérer l'utilisation des logiciels en fonctions des postes d'ordinateurs. Bien entendu, le choix d'une adresse IP pour les valeurs possibles de la variable IDIP ne concerne que la présente réalisation. Dans d'autres modes de réalisation, IDIP pourra être définie autrement.

Ainsi, selon la réalisation de la présente invention, lorsque le microcontrôleur 100 du lecteur LCL aura terminé le traitement des informations reçues et en particulier l'exécution de la fonction F_i correspondante, les résultats obtenus sont alors transportés à nouveau vers ledit ordinateur hôte à travers le réseau considéré. A réception, DRV retourne les résultats à FF_i qui les retourne au logiciel LD. Si la carte CL qui est connecté sur le lecteur LCL ne possède pas de licences d'utilisation pour le logiciel LD, alors LCL ne retournera pas de résultats mais un message indiquant à FF_i de fermer l'exécution du logiciel LD. Une fermeture d'exécution peut être donnée à FF_i dans d'autre situation décrite par la suite. Ainsi, l'exécution des fonctions F_i par le lecteur LCL crée une dépendance physique du logiciel LD avec LCL.

Ainsi, selon la réalisation de la présente invention, au début du lancement du logiciel LD, la fonction FF_0 associé à LD est exécutée. Durant l'exécution de FF_0 associé audit logiciel LD aucune autre fonction de type FF_i associée à LD ne doit être exécutée. Des informations supplémentaires seront données par la suite. FF_0 envoie eF_0, l'heure courant indiquée par l'ordinateur qui exécute FF_0, les paramètres d'exécution PARAM de la fonction F_0, la valeur S# associée au logiciel LD.

A l'arrivée des résultats calculés par le microcontrôleur 100 de LCL, FF_0 considère deux types de résultats. Le premier type concerne les résultats liés à l'exécution de F_0 (F_0 doit être une fonction compliquée de telle sorte que le logiciel LD en dépend énormément) dans le microcontrôleur 100. Dans la mesure où ces résultats ne sont des messages d'erreurs, ces résultats sont retournés au logiciel LD pour continuer l'exécution de LD. Le deuxième type concerne une heure Htop par

rapport à l'horloge dudit ordinateur exécutant le logiciel LD. Cette indication de temps correspond au prochain moment où FF_0 devra être impérativement lancé par le logiciel LD.

De plus, l'ordre dans laquelle les autres fonctions FF_i sont appelées, n'est pas prévisible, car il dépend de l'utilisation de LD.

Selon la réalisation de la présente invention, l'exécution de FF_0 est volontairement longue, de l'ordre de 1 seconde.

De plus, à l'heure Htop, si FF_0 n'est pas exécutée, alors LCL considère que la session d'exécution associée audit logiciel LD qui doit lancer FF_0 est fermée. Cette condition permet à LCL de diminuer son compteur de licences utilisées sur le réseau par rapport à LD.

Bien sûre, le dépassement du nombre de licences ne concerne pas les logiciels qui sont utilisé avec un ordinateur isolé et connecté directement par un port E/S à un LCL adapté à ce port.

De plus, lors de l'exécution des autres fonctions FF_i pour i différent de 0 par le logiciel LD, FF_i vérifie d'abord si la fonction FF_0 associée à LD n'est pas en cours d'exécution. En l'absence d'une exécution en cours de FF_0, FF_i pour i différent de 0, envoie eF_i, S#, les paramètres d'exécution de F_i vers LCL via DRV. Lorsque le lecteur LCL a terminé le traitement de eF_i accompagné de ses paramètres, les résultats sont retournés à FF_i. FF_i retourne alors ces résultats au logiciel LD. Si un message d'erreur est reçu, l'exécution du logiciel LD s'arrête. Contrairement à des systèmes de protection de logiciels qui utilisent des vérifications de codes, les logiciels protégés selon la présente invention, possèdent une protection incontournable par rapports à la difficulté de pouvoir trouver une fonction équivalente aux fonctions de F_i utilisées.

Selon la réalisation de la présente invention, compte tenu du processeur CPU1, le système d'exploitation du microcontrôleur 100 est un système multitâche afin de pouvoir traiter plusieurs utilisation de logiciels protégés selon la présente invention en même temps. La réalisation de ce système d'exploitation se réfère à des standards de systèmes multitâches existant concernant les processeurs 80386 de la société Intel. De plus, la sécurité du système de protection de logiciels selon la présente invention, repose en particulier sur l'exécution d'un seul programme principal à la fois par le processeur CPU2.

Selon la réalisation de la présente invention, à la réception complète d'un paquet d'informations, CPU1 les charge dans la mémoire DRAM 109. Dans le cas de données (correspondantes à un logiciel donné numéroté S#) du type eF_i et des paramètres associés à eF_i, la liste de test suivant est réalisée : i est égale à 0 701, eF_i correspond à un logiciel numéroté S# qui a déjà exécuté une fois FF_0 avec succès 702, l'heure indiquée par l'horloge 104 est égale à l'heure Htop (valeur précédemment définie correspondant au logiciel numéroté S# qui a dans cette condition déjà exécuté une fois FF_0 avec succès) exprimée par rapport à l'heure de l'horloge 104 avec une erreur de plus ou moins deux secondes (test 703 ou 705), le nombre (noté NL.S#) courant de licences utilisées (correspondant au logiciel numéroté S#) augmenté de 1 est strictement supérieur à la valeur L#.S# du fichier Fich.S# (correspondant au logiciel numéroté S#) fourni par la carte CL (test 704). Les éléments de cette liste de tests sont notés respectivement test1, test2, test3,

test4. Le chiffre associé à ces tests indique l'ordre de ces tests dans les conditions de leurs réalisations. En référence à la figure 12, le test test1 701 correspond au début de l'arbre d'analyse qui est effectué donc en considérant un logiciel donné numéroté S#, et un identifiant unique IDIP qui est le numéro adresse IP de chaque ordinateur sur le réseau considéré. Ce numéro permet ainsi d'associer chaque session d'exécution de logiciels par rapport à un ordinateur donné dudit réseau. Selon des variantes de cette réalisation, d'autre identifiant unique peuvent être associé directement à une session d'exécution d'un logiciel donné par l'emploi d'un identifiant de processus combiné avec l'adresse réseau de l'ordinateur où se trouve ce processus. Ainsi, la présente réalisation considère l'identifiant unique associé à une adresse IP comme un exemple de moyen possible pour identifier un ordinateur sur ledit réseau. Ainsi, au début de l'arbre d'analyse de la figure 12, les informations S# et IP (IDIP) sont supposées fournies par la fonction FF_i avec le driver DRV. Ces deux informations sont envoyées vers ledit lecteur LCL dans le paquet d'informations PARAM précédemment décrit.

Selon la réalisation de la présente invention, par rapport au test test4 704, si le résultat est une valeur fausse au sens booléen, alors la nouvelle valeur de NL.S# 709 est celle de l'ancienne augmentée de 1, à condition que le test test2 702 soit faux et que le test1 701 soit vrai.

Par rapport à une valeur vraie du test test3 703, et à condition d'une valeur vraie des tests test1 701 et test2 702, CPU1 effectue un lecteur de l'heure courante sur l'horloge interne 104, pour calculer la nouvelle valeur de Htop exprimée par rapport à l'heure de l'horloge associée à l'ordinateur exécutant le logiciel S# dans les conditions de ces tests et le contexte de ces conditions. La nouvelle valeur de Htop est à égale l'ancienne augmentée de 5 minutes (par exemple), selon la réalisation de la présente invention. Cette valeur de 5 minutes est ajustée de telle sorte que deux ou plusieurs logiciels ayant le même numéro S# et tournant sur des ordinateurs différents, ne puissent pas exécuter la fonction FF_0 correspondante en même temps (il faut tenir compte de ladite erreur de plus ou moins deux secondes). Cette valeur de 5 minutes peut donc être remplacée, par tout autre valeur en fonction de la phrase précédente. A la fin du test test3, une exécution 706 de la fonction F_0 a lieu.

Par rapport à une valeur fausse du test test3 703, et à condition d'une valeur vraie des tests test1 701 et test2 702, un message d'erreur 707 est alors retourné à la session d'exécution du logiciel correspondant à l'exécution de la fonction FF_0 dans les conditions de ces tests et le contexte de ces conditions. De plus, CPU1 diminue de 1 point le compteur NL.S# associé aux logiciels au numéro de série S#. Le système d'exploitation du microcontrôleur 100 dans le cadre de la protection de plusieurs logiciels en même temps, gère une liste d'objets référencés par les différents identifiants IDIP correspondant à tous les ordinateurs du réseau qui exécutent un logiciel protégé selon la présente invention. Cette liste d'objets est établie à partir des informations de PARAM. Les champs de chacun de ces objets servent à enregistrer les numéros de série S# des logiciels numérotés S# qui sont exécutés sur l'ordinateur dont l'identifiant IDIP correspond à ladite référence IDIP de ces objets. Cette liste d'objet permet la gestion de l'utilisation des logiciels

protégés en fonction des ordinateurs. Lorsque NL.S# est diminué de 1 point par rapport un ordinateur identifié par IDIP et qui exécute le logiciel numéroté S# qui a provoqué cette diminution, le champ de l'objet IDIP correspondant qui contient la valeur de S# est effacé. Cette gestion permet à ce que une nouvelle exécution d'un logiciel numéroté S# puisse avoir lieu. Il faut aussi noter que la valeur de NL.S# est le total desdits objets possédant un champ identique à la valeur de S# dans la notation de NL.S#. Ainsi, en contrepartie lorsqu'une augmentation de 1 point de la valeur de NL.S# a lieu, un nouveau champ dans l'objet IDIP de ladite liste d'objets est créé. L'objet IDIP correspond à l'ordinateur numéroté IDIP qui exécute la fonction FF_i du logiciel S#. Ledit nouveau champ prend alors la valeur de S#.

Par rapport à une valeur vraie du test test4 704 et d'une valeur fausse du test test2 702 et d'une valeur vraie du test test1 701, un message d'erreur 708 est alors retourné à la session d'exécution du logiciel correspondant à l'exécution de la fonction FF_0 dans les conditions de ces tests et le contexte de ces conditions.

Par rapport à une valeur vraie du test test3 705, et à condition d'une valeur fausse du test test1 701, CPU1 diminue de 1 point le compteur NL.S# associé aux logiciels numérotés S#. Ladite liste d'objets IDIP est actualisée en conséquence. Un message d'erreur 710 est alors retourné à la session d'exécution du logiciel correspondant à l'exécution de la fonction FF_0 dans les conditions de ces tests et le contexte de ces conditions.

En référence à la figure 12, pour une valeur vraie du produit (test1 701 et test2 702 et test3 703) dans l'ordre de leur réalisation, pour une valeur fausse du produit (test1 701 et test2 702 et test4 704) dans l'ordre de leur réalisation, et pour une valeur fausse du produit (test1 701 et test3 705) dans l'ordre de leur réalisation, le résultat conduit à un traitement de eF_i par CPU1.

Bien entendu, l'ensemble de ces tests est défini par rapport à la réalisation de la présente invention pour permettre les fonctionnalités de protection de plusieurs logiciels en même temps par un seul lecteur LCL. L'arbre d'analyse de la figure 12 est simplifié au maximum par soucis de clarté de compréhension. Selon des modes particuliers de réalisation, les conditions des tests et le contexte de ces tests peuvent varier.

En cas d'utilisation intensive, une file d'attente est créée pour faire exécuter une à une les fonctions F_i par le processeur CPU2. Afin de diminuer le temps d'attente, une condition de durée autorisée pour l'exécution d'une fonction F_i donnée peut être fixée, par exemple 1/100 de secondes valeur définie par rapport à la vitesse de calcul du processeur CPU2. Cette condition de durée devra être respectée par le développeur de logiciels protégés selon la présente invention.

Selon la réalisation de la présente invention, les informations d'utilisation d'un logiciel donné numéroté S# sont fournies par la carte CL. Ainsi, lorsqu'il est nécessaire par rapport à l'arbre d'analyse, notamment à l'étape 704, le microcontrôleur 100 effectue une requête auprès de la carte CL. Ainsi, CPU1 effectue dans l'objectif d'exécuter F_i, une requête à propos du logiciel numéroté S#, auprès de la carte CL connecté dans le contexte de cette requête. Dans les conditions de succès de la procédure d'authentification précédemment définie entre LCL et CL, et dans les conditions de

la possessions du fichiers Fich.S# associé au logiciel numéroté S# du contexte de cette requête, le microcontrôleur 400 de la carte CL retourne au LCL sous une forme codée par la clé kCS, le fichier Fich.S#. Le microcontrôleur 100 met alors à jour les champs suivants : DR.S# (dernière utilisation) remplacée par la date courante, DU.S# recalculé, Nombre d'exécution du logiciel. DR.S# est mise à jour par rapport à la date et l'heure de la première exécution de la fonction FF_0 par rapport au dernier lancement d'une exécution du logiciel S# associé à cette fonction. DU.S# est recalculée avec la valeur dynamique de Htop et l'heure courante indiquée par l'horloge 104 au moment où l'exécution d'une fonction FF_i correspondante a lieu. Le champ « Nombre d'exécution du logiciel » est calculé par rapport à la dernière exécution du logiciel associé au contexte de ce calcul. Il est donc augmenté d'un point à chaque nouvelle exécution. L'ensemble de ces tests est optimisé en fonction du flux des requêtes des différentes fonctions FF_i exécutées par les logiciels correspondants sur les différents ordinateurs connectés sur le réseau avec le lecteur LCL considéré dans le contexte de ces mises à jour d'informations.

Selon la réalisation de la présente invention, les valeurs de L#.S#, kEL.S# et kX.S# (d'autres champs du fichier concernés peuvent être retenus par CPU1 selon les besoins) sont mémorisées dans la DRAM 109. Fich.S# modifié est ensuite retourné à CL sous une forme codée par kCS. Afin d'empêcher des actes de piraterie, si la carte CL est enlevée pendant que le lecteur LCL utilise la carte CL pour effectuer des opérations concernant la présente invention, le lecteur LCL met fin à toutes les sessions d'exécution de logiciels protégés selon la présente invention, en leur retournant un message d'erreur. Selon des modes particuliers de réalisation, les appareils selon la présente invention, peuvent ne pas retourner ainsi un message d'erreurs, dans la mesure où des fichiers Fich.S# peuvent être stockés en permanence dans le lecteur LCL de la même manière que celle employée avec la carte CL.

Ainsi, selon la réalisation de la présente invention, la valeur L#.S# empêche de dépasser le nombre de licences d'utilisations d'un logiciel protégé selon la présente invention.

Selon la réalisation de la présente invention, pour le cas i=0, la clé kEL.S# fournie par le fichier Fich.S# est utilisée pour décoder eF_0. CPU1 obtient ainsi ledit fichier de conditions d'utilisation du logiciel associé à ce eF_0. CPU1 compare la valeur de ces différentes informations par rapport au fichier Fich.S# associé. A titre explicatif, si l'utilisation est fixée par rapport à une date d'expiration, ledit fichier de conditions d'utilisation renseigne cette valeur limite par son champ correspondant. Selon des modes particuliers de réalisation, ledit fichier de conditions d'utilisation peut ne pas comporter tout ou partie des champs suivants : licences permanentes, durée d'utilisation, l'utilisation expire fin, nombres d'exécutions.

Selon la réalisation de la présente invention, après le succès de la comparaison des données fournies par le fichier Fich.S# avec le fichier de conditions (limites) d'utilisation décodée par la clé kEL.S# et associée à eF_0, et toujours pour i=0, CPU1 récupère aussi avec la clé kEL.S# les « byte code » JAVA (code d'instruction JAVA) de la fonction F_0.

Selon la réalisation de la présente invention, pour i différent de 0, les données codées eF_i sont décodées à l'aide de la clé kX.S# renseignée par le Fich.S# correspondant. CPU1 récupère alors les « byte code » JAVA (code d'instruction correspondant à la machine virtuelle JAVA) de la fonction F_i.

Ces codes d'instruction sont chargés alors par l'intermédiaire de l'interface 106 directement dans la DRAM 110 grâce au contrôleur DMA 107. Par l'intermédiaire du contrôleur du CPU2, CPU1 donne la main au CPU2 (processeur PicoJava) pour l'exécution de F_i dont les paramètres d'exécution ont été chargés au préalable. Le watchdog 108 surveille au bon fonctionnement du CPU2. A la fin de l'exécution de F_i par CPU2, les résultats sont récupérés par CPU1 et retournés vers l'ordinateur qui a envoyé les données eF_i.

Dans les contextes d'utilisation 30 et 20, un lecteur LCL peut être utilisé avec un ordinateur personnel grâce à une connexion directe entre leur port E/S USB respectif. Pour ces deux contextes la protection de logiciels par le lecteur LCL est une version simplifiée des fonctionnalités en réseau. Par conséquent, la présente description n'apportera pas de renseignements supplémentaires.

Selon des modes particuliers de réalisation, le champ « durée d'utilisation » du fichier de conditions d'utilisation peut être utilisé pour caractériser l'utilisation de logiciels de démonstrations. Ainsi, le fichier de conditions d'utilisations fixe en particulier des conditions d'utilisation limite. Le fichier d'autorisation d'utilisations renseigne la « quantité » d'utilisation en cours. Les deux fichiers combinés permettent selon la réalisation de la présente invention, de protéger un logiciel contre le non respect de ses conditions d'utilisations.

De plus, le format du fichier de conditions d'utilisation est surtout intéressant pour développer des applications de démonstrations dont l'utilisation est limitée. Dans ce contexte, une procédure de protection particulière peut être utilisée pour que la création de logiciels protégés selon la présente invention, puisse être effectué sans passer par l'intermédiaire du serveur aSVR. Cette fonctionnalité est intéressante pour permettre notamment de décentraliser la protection de logiciels selon la présente invention. De plus, pour lancer une telle procédure de protection, le choix sera effectué à l'aide du programme PGM.

Par rapport à la procédure de protection de logiciels précédemment décrite, les fonctions F_i sont maintenant codées avec une clé secrète kLi de ladite liste $\{kL_{i+1},kL_{i+2},kL_{i+3},\ldots,kL_{i+m}\}$, qui correspond à la semaine courante pendant laquelle cette procédure spéciale de protection de logiciels est lancée. L'usage de ces clés est intéressant uniquement pour les logiciels dont l'utilisation est limitée dans le temps à cause de la définition de ces clés kLi. Bien sûre, le lecteur LCL qui effectuera cette procédure ne communiquera pas la valeur de kLi par rapport à sa définition. Pour être distribué, le logiciel ainsi protégé devra être accompagné des indications sur la semaine pendant laquelle cette procédure particulière de protection a été effectuée. Ainsi, lorsqu'un utilisateur désire exécuter un logiciel protégé de cette manière (par la clé kLi), il n'aura pas besoins de contacter le serveur aSVR, car toutes les informations s'y trouvent sur le média d'enregistrement. Cependant l'utilisation du logiciel reste dépendante d'un lecteur LCL compte

tenu du cryptage. Ainsi, l'utilisateur pourra disposer immédiatement du logiciel, mais ne pourra pas dépasser les conditions d'utilisations fixées dans le fichier de conditions d'utilisation. La réalisation de la présente invention empêche une réutilisation d'un logiciel dont l'utilisation est limitée dans le temps. Des explications seront données par la suite.

De plus, la présente invention concerne une méthode rendant l'utilisation des appareils selon la présente invention, transparente du point de vue informatique par l'intermédiaire d'un programme noté PGM qui a déjà été abordé. Le programme PGM est développé de telle sorte qu'il puisse permettre à un utilisateur des appareils selon la présente d'effectuer des opérations nécessitant une interactivité avec ces appareils. Son utilisation est généralement sous-entendue dans la présente description. Il a aussi pour rôle de permettre à un LCL donné de se connecter à un système informatique distant en utilisant les ressources de communication de l'ordinateur hôte et du système informatique de cet ordinateur.

Le programme PGM est utilisé en parallèle avec un programme driver DRV. Ce driver DRV constitue une couche de communication entre PGM et un lecteur LCL donné. Il assure la transparence de l'utilisation du lecteur LCL. L'adjonction de ces deux éléments dans un ordinateur donné est illustré sur la figure 2. Ces deux programmes remplissent toutes les fonctionnalités décrites précédemment et par la suite. Une convention est adoptée sur les procédures d'interrogation d'un LCL donné afin de permettre à ce lecteur LCL de reconnaître et d'exécuter des commandes qui sont intégrés au système d'exploitation de son microcontrôleur 100. Ces commandes sont définies lors de la construction des lecteurs LCL.

En retour, selon la présente invention, PGM peut aussi interpréter les informations provenant des commandes envoyées par LCL. Ces commandes sont essentiellement, selon la réalisation de la présente invention, des instructions pour permettre à un LCL d'accéder à un système distant.

Ainsi, selon la convention adoptée pour communiquer avec un lecteur LCL donné, des commandes sont définies par rapport à la possibilité pour les lecteurs LCL, de se connecter à aSVR par l'intermédiaire des ressources disponibles de communication réseau pour permettre par exemple une mise à jour du système informatique interne des appareils selon la présente invention ou encore la mise à l'heure de l'horloge interne 104 en cas de dysfonctionnement (la pile 103 est épuisée).

Selon le contexte d'utilisation 20 indiqué sur la figure 1, un récepteur hertzien numérique 22 est connecté sur le bus externe 114 du microcontrôleur 100, pour permettre à un lecteur LCL donné de recevoir des informations directement de l'organisme qui gère le serveur aSVR. Bien entendu, ce récepteur sera intégré dans le boîtier retenu pour l'utilisation du lecteur LCL. Il est ainsi possible d'envoyer des informations à tous les lecteurs LCL en service de manière générale et/ou spécifique. De plus, la faible consommation de ce type de récepteur par rapport à un fonctionnement avec des piles électriques, permet de les laisser en fonctionnement permanent, même si le lecteur LCL est éteint. Selon une variante d'utilisation d'alimentation électrique, des piles rechargeables peuvent être employées avec le radiorécepteur 22 indépendamment de l'alimentation électrique du LCL

(fournit éventuellement par le réseau électrique local). Bien entendu, le récepteur dispose de sa propre mémoire pour permettre de conserver les données reçues de l'émetteur 13 lorsque le lecteur LCL est éteint. Ainsi, le serveur aSVR peut envoyer des informations comme par exemple la mise à jour du système d'exploitation du lecteur LCL et/ou de la carte CL par l'intermédiaire d'un émetteur 13.

Selon une variante du fonctionnement des lecteurs LCL, une condition de durée peut être ajouter en plus des informations concernant les dates de mise en service et de fin d'utilisation (DB.d et DE.D). Cette condition est relative à l'utilisation de LCL avec le radiorécepteur 22. Ainsi, un lecteur LCL qui n'aurait reçu aucunes informations provenant de l'émetteur 20, refusera de fonctionner au moment de sa mise en marche par un utilisateur. Une procédure de connexion sur aSVR par l'intermédiaire du programme PGM devra être effectuée afin de récupérer les informations que ce lecteur LCL aurait pu manquer pour cause de mauvaise réception radio. Cette récupération se fait bien sûre par l'intermédiaire de la clé kT.d pour sécuriser la communication entre aSVR et le lecteur LCL en question.

De plus, afin d'envoyer de l'émetteur 13 des informations de manière sécurisée, les clés secrètes kLi précédemment décrites sont utilisées pour coder ces informations à envoyer par radio aux lecteurs LCL. On note ces informations MR. De plus, kLi est choisie de telle sorte qu'elle correspond avec la semaine où les informations partent de l'émetteur 13. On note eMR la forme codée de MR par kLi. En choisissant d'émettre les informations MR à partir du lundi qui suit la semaine (commençant un lundi et se terminant fin dimanche) où ces informations MR ont été définies, les mêmes informations sont envoyées en répétition suivant un intervalle donné durant toute la semaine. Ceci permet de s'assurer qu'elles ont bien été reçues, et d'éviter trop de connexion de LCL vers aSVR.

Cette variante de la présente invention présente beaucoup d'avantage, car elle permet dans un premier temps de retourner les fichiers Fich.S# de manière sécurisée vers le lecteur LCL correspondant à l'acheteur du logiciel associé à Fich.S#. Bien entendu, l'émetteur 13 peut envoyer de manière spécifique des informations vers un récepteur radio 22 donné. Cette variante permet un achat sans connexion informatique, mais par l'utilisateur directement au téléphone avec un accueil humain. Cette variante permet une complète séparation de la distribution du média d'enregistrement contenant un logiciel donné; de la vente des licences d'utilisation de ce logiciel donné.

Dans un deuxième temps, cette variante permet d'envoyer des informations concernant la perte d'un appareil. Compte tenu des capacités de stockage des modules de « Disque Flash », un circuit DiskOnChip non illustré de la société Msystems est connecté avec un contrôleur éventuel sur le bus externe 114 du microcontrôleur 100 pour permettre au microcontrôleur 100 de disposer d'un disque de stockage. Un DiskOnChip de 12 mégaoctets est choisit selon la réalisation de la présente variante de l'invention. Selon d'autres variantes, une carte Disque Flash PCMCIA peut

être employée à la place du DiskOnChip. Ainsi, à réception des informations eMR, le microcontrôleur 100 décode eMR à l'aide de la clé kLi de la semaine courante.

Ainsi, pour désactiver un appareil CL et/ou LCL donnés par rapport à leur utilisation, une information correspondant à son numéro de série peut être jointe. Ce numéro de série est alors sauvegardé sur le circuit DiskOnChip dans un fichier noté ANNUL qui sert à stocker tous les numéros de séries des appareils selon la présente invention, qui ne doivent plus être utilisés. Selon des modes particuliers de réalisation, par rapport à la sécurité nécessaire contre des modifications, ledit numéro peut ne pas être codé lors de son émission de l'émetteur 13.

Selon la présente invention, un procédé informatique d'authentification et de signature par LCL est effectué sur le fichier ANNUL. La signature électronique et les informations de l'authentification sont stockées dans la mémoire interne 111 du microcontrôleur 100. Ainsi, le microcontrôleur 100 à chaque démarrage vérifie si le fichier ANNUL n'a pas été remplacé par un autre fichier au même format ou modifié par une opération non autorisée.

Ce fichier ANNUL est alors utilisé lors de la procédure d'authentification entre une carte CL donnée et un lecteur LCL donnée par rapport à la procédure précédemment décrite. Si CL présente un ID.c qui est référencé dans le fichier ANNUL, LCL rejette alors la carte CL en question. De plus, au démarrage de LCL et/ou à réception d'une information MR, le microcontrôleur 100 du LCL, vérifie si son propre ID.d n'est pas référencé dans le contenu de MR et/ou ANNUL. Si le cas se présente, le microcontrôleur 100 se met hors service en détruisant le contenu de sa mémoire interne.

Ainsi, compte tenu de la réalisation de la présente variante, un appareil CL ou LCL peut être mis hors service dans un maximum de 1 semaine.

De plus, selon la réalisation de la présente invention, les appareils ont une utilisation d'au maximum 4 ans. Cette durée peut tout à fait se ramener à 2 ans. Dans le contexte de cette durée, en considérant une possibilité de pertes des appareils selon la présente invention, avec un volume de 1 million de pertes d'appareils en 2 deux ans (les pertes volontaires seraient empêchées par le prix d'achat d'un nouvel appareil en cas de perte). Ce volume peut sembler exagérer. Compte tenu des capacités des méthodes de compressions (un taux de 50%), et compte tenu de la taille de 128 bits d'un numéro de série ID, il faudrait donc environ et sans compression 15 mégaoctets, 8 mégaoctets avec compression d'espace de stockage de données. Sur une durée d'utilisation de deux ans des appareils selon la présente invention, la capacité de la DiskOnChip selon la présente variante suffit. Pour le cas de durée d'utilisation plus longue, des capacités de stockage plus grandes peuvent être prises compte tenu de la capacité des modules DiskOnChip dans l'état actuel de l'art.

De plus, ledit radiorécepteur numérique, compte tenu de sa capacité de recevoir des informations qui le concerne uniquement, lors de l'achat d'une licence de logiciel, l'utilisateur peut recevoir par radio le fichier Fich.S# (décrite précédemment) correspondant à son achat de licences du logiciel numéroté S#. Cette fonctionnalité peut avoir un grand impact au niveau commercial (l'achat de logiciel peut être effectué partout sur la planète sans connexion). Bien sûre, Fich.S# est

envoyé sous forme codée avec la clé kLi de la semaine courante. Selon des variations de réalisations, on peut utiliser une clé de type kT.d pour les opérations d'achat.

La possibilité de faire rejeter l'utilisation d'un appareil selon la présente invention par le reste des appareils selon la présente invention, permet de donner aux utilisateurs la possibilité de récupérer une partie du contenu de leur carte CL perdue.

Ainsi, en tenant compte du format de fichier Fich.S#, la taille de ce fichier peut être ramenée à 66 octets dans un fichier nommé rFich.S#, en conservant uniquement les champs suivants : S# du logiciel, ID.c, L#.S#, kEL.S#, kX.S#. En considérant uniquement les logiciels dont l'utilisateur possède une licence d'utilisation permanente, rFich.S# suffit pour définir l'utilisation de ces logiciels protégés selon la présente invention. Ainsi avec un module de mémoire de 64 kilooctets, on peut stocker au moins 990 licences de logiciels différents dont l'utilisation peut être définie par rFich.S#. Selon cette nouvelle variante, l'organisme qui gère aSVR, fournit lors de l'achat d'une carte CL une carte à puces (SmartCard) notée SC, et pouvant sécuriser des données en lecture et en modification. SC n'est pas illustrée. SC comporte un microcontrôleur intégrant sur une seule pastille de silicium, un processeur, un module mémoire Flash de 64 kilooctets, de la mémoire DRAM et OTPEPROM. Les accès en mémoire sont contrôlés par le processeur du microcontrôleur de SC. Cette carte est une carte à puces sécurisée. Cette carte à puces est utilisée à chaque fois que l'utilisateur acquière légalement une nouvelle licence pour une ou des utilisation(s) permanente(s) d'un logiciel donné protégé selon la présente invention. Lors de cette acquisition, cette carte est insérée dans le lecteur de cartes à puces qui communique avec le microcontrôleur 100 du lecteur LCL par le moyen du contrôleur de cartes à puces 153. Le fichier rFich.S# sera alors copié dans le module de mémoires Flash de 64 kilooctets du microcontrôleur de la carte à puces. Bien entendu, la carte à puces SC possède en interne une clé secrète kLi de la même manière que la carte CL. Cette clé est utilisée lors d'une procédure d'identification similaire à celle qui a lieu entre le lecteur LCL et la carte CL. Des modifications du contenu de la carte SC ne peuvent être effectuées uniquement par un lecteur LCL connecté à une carte CL associée. Ainsi, pour cette variante de la réalisation de l'invention, rFich.S# est stocké sous une forme codée, notée erFich.S#, par la clé kS.c de la carte CL. De plus, compte tenu des propriétés de sécurité des cartes à puces (SmartCard), erFich.S# et ainsi protégé contre la modification ou la lecture non autorisée. De toutes manière cette information est protégé. De plus, par l'emploi de la clé kS pour codée rFich.S#, cette carte ne peut être utilisée qu'avec la carte CL d'où les fichiers rFich.S# proviennent. Bien sûre, si l'utilisateur a acheté des licences d'un logiciel S# en plus par rapport à celles qu'il a déjà dans sa carte CL, LCL copie alors erFich.S# et l'envoie vers la carte CL correspondant. Les champs de rFich.S# sont mis à jour par rapport au nombre de licences nouvellement acquises. Le nouveau fichier erFich.S# obtenu remplace ensuite l'ancien dans la mémoire interne du microcontrôleur de la carte SC.

Ainsi en cas de perte, la sauvegarde effectuée sur la carte à puces SC peut être récupérer en deux étapes : achat d'une nouvelle carte CL, connexion vers le serveur aSVR par l'intermédiaire du programme PGM.

Lors de la connexion sur le serveur, l'utilisateur communique par l'intermédiaire de son programme PGM, le numéro de série ID.c de son ancienne carte CL (ID.c est une donnée publique non modifiable : il est affiché en clair sur le boîtier 60 de chaque CL). Ensuite, l'utilisateur communique le numéro de série de sa nouvelle carte CL. En échange, aSVR retourne une donnée qui est la forme codée de la clé secrète kS de la carte CL perdu. Cette clé est codée par la clé kT.d du lecteur LCL sur lequel la nouvelle carte CL est connectée. L'acquisition de la clé kS de la carte CL perdue permet de récupérer ainsi le contenu des fichiers rFich.S#.

Selon la variante de l'invention utilisant le radiorécepteur, le numéro ID.c de la carte CL perdue peut être communiqué oralement par téléphone. LCL reçoit alors par l'intermédiaire de son radiorécepteur numérique et sous une forme codée par la clé kLi de la semaine courante, la clé kS de la carte CL perdue et correspondant audit numéro ID.c de la carte perdue. L'acquisition de la clé kS de la carte CL perdue permet de récupérer ainsi le contenu des fichiers rFich.S#.

De l'autre coté, aSVR lance une procédure pour désactiver l'utilisation de la carte perdue en envoyant selon la variante de l'invention précédemment décrite, le numéro ID.c de la carte perdue à tous les lecteurs LCL.

Selon une autre variante de l'invention, le stockage des fichiers Fich.S# peut être effectué par le lecteur LCL en suivant les conditions de sécurités similaires au fonctionnement d'une carte CL au niveau du stockage de ces fichiers. Les fichiers Fich.S# seront alors stockés sur média d'enregistrement externe prévu pour ce stockage. On peut utiliser par exemple une DiskOnChip. Les fichiers stockés sur ce support sont protégés par la clé kS.d du lecteur LCL correspond. Dans cette variante, la clé kS.d est une clé secrète inscrite dans la mémoire interne 111 lors de la programmation en usine du microcontrôleur 100. Ainsi lorsqu'une ou des licence(s) de logiciels sont déplacés vers le lecteur LCL, l'accès aux logiciels protégés selon la présente invention, et associés à ce lecteur LCL peut être réalisée indépendamment de la présence d'une carte CL. Ceci permet une utilisation par toutes les personnes pouvant accéder à l'ordinateur sur lequel ledit lecteur LCL est connecté. Bien entendu, lors d'un déplacement de licences de logiciels d'une carte CL, les informations seront mises à jour dans la carte à puces SC en obligeant la connexion de la carte à puces SC correspondant à cette carte CL sur le lecteur adapté du lecteur LCL. Par exemple lorsque deux licences d'un logiciel numéroté S# sont déplacée de la carte CL vers un lecteur LCL, le contenu du fichier eFich.S# de la carte CL et le contenu du fichier erFich.S# de la cartes à puces SC seront modifiés en conséquence afin d'écrire un fichier d'utilisations de logiciels au niveau du lecteur LCL. Dans cette nouvelle fonctionnalité du lecteur LCL, une carte à puces du même type que SC devra être associée à chaque lecteur LCL pour permettre une sauvegarde des fichiers Fich.S# relatifs à des licences d'utilisations permanentes de logiciels. Ainsi, cette carte à puces devra être insérée immédiatement après les déplacements des licences d'utilisations de la carte CL vers le lecteur LCL pour valider le transfert. Compte tenu du fait que les autorisations d'utilisations de logiciels copiées dans le lecteur LCL peuvent être transférer par une opération inverse à celle qui vient d'être décrite vers une nouvelle carte CL, il est donc possible de déplacer une autorisation

d'utilisation d'un logiciel protégé selon la présente invention d'une carte CL vers une autre carte CL. Les cartes à puces SC respectives de ces deux cartes CL seront bien entendu mises à jour automatiquement.

De plus, par rapport à ladite possibilité de changer de carte CompactFlash sur une carte CL, et par rapport à des autorisations d'utilisation de logiciels relativement à une utilisation limitée, un fichier eTPS est présent sur chaque carte CompactFlash qui est utilisée par une carte CL donnée et numérotée ID.c. Ce fichier devra impérativement être présent sur toutes les cartes CompactFlash utilisées par ladite carte CL. Autrement, la carte CL ne fonctionne pas. De plus, eTPS est la forme codée du fichier TPS qui contient en première ligne le numéro ID.c, puis tous les numéros de série de logiciels S# acquis selon la présente invention, par l'utilisateur de cette carte CL. Ainsi, un utilisateur ne pourra contourner les limites d'utilisations d'un logiciel d'utilisation limitée par rapport par exemple au temps ou au nombre d'exécutions, en changeant de carte CompactFlash. Cette restriction s'applique selon la réalisation de la présente invention, par exemple sur les logiciels gratuits.

Ainsi selon des variantes évidentes de réalisation qui ne seront pas décrits (cette description n'apporte rien à la compréhension et à la réalisation de la présente invention), par rapport à sa capacité de protection d'informations contre toutes modifications, la carte CL peut servir au stockage d'informations public non modifiable tel que l'identité d'une personne. Ces informations pourront être consultées par l'intermédiaire de l'interface utilisateur représentée par le programme PGM. La carte CL peut, en effet compte tenu de sa grande capacité de stockage permettre de stocker des programmes et/ou des compteurs relatifs à une valeur donnée, de manière sécurisée contre des modifications et/ou lectures non autorisées selon un critère donné. De l'autre côté, LCL offre un moyen sécurisé pour permettre l'exécution de ces programmes supplémentaires et/ou le traitement de ces compteurs relatifs à une valeur donnée. Le microcontrôleur 100 est physiquement protégé contre des programmes dits virus informatiques. On peut ainsi envisager de définir le champ Misc des fichiers Fich.S# par des codes de programmes exécutables par le processeur CPU2 du microcontrôleur 100 ou par des compteurs de fidélité représentant le nombre de licences de logiciels achetés par l'utilisateur à un concepteur de logiciels donnés, afin de permettre des opérations commerciales correspondantes. Les codes de programmes éventuellement ajoutés dans le champ Misc peuvent permettre de modifier le comportement desdites fonctions F_i lors de leur exécution par le lecteur LCL, afin de rendre quasiment impossible le piratage du logiciel correspond en essayant de remonter à la fonction F_i par une surveillance des entrées et sorties de données au niveau des appels des fonctions FF_i. Il est à rappeler que seule la forme codée eF_i des fonctions F_i sont accessibles par l'utilisateur.

Selon la présente invention, l'emploi d'un circuit intégré (le microcontrôleur 100) physiquement et logiquement protégé contre des attaques de virus informatiques et contre les lectures et/ou des modifications de données contenues dans le circuit, permet un très haut niveau de sécurité de protection de logiciels. Compte tenu des périphériques de communications que peut

utiliser un tel circuit, les appareils selon la présente invention, n'apparaisse plus comme un appareil prohibitif de surveillance, mais un véritable outil participant activement à l'utilisation des logiciels notamment par le fait que la présente invention permet de récupérer en sécurité des licences perdues. En effet, la présente invention permet de rendre un outil de protection de logiciels, d'aspect souvent décoratif en raison du temps où il est effectivement utilisé, en un outil que l'utilisateur peut utiliser dans sa vie quotidienne par sa capacité de sécuriser le stockage de données et l'exécution de programmes. Ainsi la présente invention est un nouvel outil de protection de logiciel qui permet d'une part la distribution des logiciels indépendamment de la vente de leur droit d'utilisation, et d'autre part un développement libre des logiciels protégés selon la présente invention. Selon la présente invention, cette séparation a une grande conséquence sur le coût nécessaire pour protéger un logiciel. En effet, l'utilisation d'un seul appareil selon la présente invention, pour la protection de plusieurs logiciels indépendamment des concepteurs de logiciels, permet de distribuer le coût d'un appareil selon la présente invention sur tous les concepteurs de logiciels, de sorte que le prix d'un seul appareil selon la présente invention, devienne faible et abordable pour l'utilisateur.

De plus, ladite séparation permet selon la présente invention, la vente de moyens de protections de logiciels indépendamment des logiciels qui utilisent ces moyens pour la protection. La puissance de la présente invention au niveau des sécurités utilisées, permet compte tenu de sa vente séparée du produit logiciel, une exploitation commerciale liée à d'autres opérations. Ces opérations peuvent être des opérations qui consiste à présenter des informations confidentielles que l'utilisateur ne peut modifier ou falsifier. Elle peut servir donc à un outil permettant des accès à un système donné. Ainsi, la présente invention est un moyen de protection de logiciels qui permet des fonctionnalités parallèles à son utilisation. Ces fonctionnalités auront pour conséquence la baisse du coût des appareils selon la présente invention et la baisse du coût de protection d'un logiciel donné. La protection de logiciel selon la présente invention, devient par conséquent un système intéressant pour les petites et les grandes productions de logiciels. Les appareils selon la présente invention sont susceptibles d'une industrialisation par rapport au monde de l'industrie des logiciels et de leur protection.

Bien entendu, l'invention n'est pas limitée aux modes de réalisation qui viennent d'être décrits et représentés. On pourra y apporter de nombreuses modifications de détail sans sortir pour cela du cadre de l'invention.

Revendication

1. Système pour la protection simultanée de plusieurs logiciels provenant de différents concepteurs de logiciels contre le non respect des conditions d'utilisation fixées par ces concepteurs de logiciels, caractérisés en ce qu'il comprend en combinaison :

- un lecteur (LCL) comprenant au moins un périphérique de communication (réseau, port E/S) créant une couche de communication supérieure permettant l'échange de données avec les logiciels protégés, un microcontrôleur programmable une seule fois (100) qui intègre sur une seule entité électronique deux parties (130, 120) séparées par une interface (106)
- un appareil portatif (CL) de type carte destiné au stockage d'un grand nombre d'autorisations d'utilisation de logiciel protégé comportant un module d'enregistrement amovible de forte capacité de stockage, et un microcontrôleur (400) sécurisé contre toutes intrusions non autorisées dans ses circuits internes.

2. Système selon la revendication 1 caractérisé en ce que la partie (130) du microcontrôleur (100) comprend au moins un module de mémoires non volatiles (111), un module de mémoires volatiles (109), un port série E/S (151), une horloge interne en temps réel (104) et un processeur maître (CPU1).

3. Système selon l'une quelconque des revendications précédentes caractérisée en ce que la partie (120) du microcontrôleur (100) comprend au moins un module de mémoires volatiles (110) et un processeur esclave (CPU2).

4. Système selon l'une quelconque des revendications précédentes caractérisée en ce que le microcontrôleur (100) sécurise physiquement et logiquement d'une part son espace mémoire interne contre la lecture et/ou la modification non autorisé, et d'autre part l'exécution de programmes au sein de la mémoire (110) vis à vis de la possibilité pour un programme donné, exécuté dans cet espace mémoire d'extraire des données du microcontrôleur (100) présentes avant l'exécution de ce programme.

5. Système selon l'une quelconque des revendications précédentes caractérisé en ce que le microcontrôleur (400) comporte au moins un module de mémoires OTPEPROM (407) ou équivalent, un module de mémoires dynamiques DRAM (107) ou équivalent, et un processeur (CPU).

6. Système selon l'une quelconque des revendications précédentes caractérisé en ce que le lecteur comprend un récepteur hertzien pour permettre des opérations d'achat hors ligne de droits d'utilisation de logiciel protégé, de mise à jour des microcontrôleurs (100, 400), ou d'administration du lecteur (LCL) et du dispositif portatif (CL).

7. Systèmes selon l'une quelconque des revendications précédentes caractérisé en ce que le lecteur comprend un périphérique de communication pour la connexion et un système central distant pour permettre des opérations d'achat de droits d'utilisation de logiciel protégé, de mise à jour des microcontrôleur (100, 400), ou d'administration du lecteur (LCL) et du dispositif portatif

(CL).

8. Système selon l'une quelconque des revendications précédentes caractérisé en ce que le microcontrôleur est sécurisé contre toute attaque physique et/ou logique et mémorise au moins une série de codes et de clés numériques, utilisés pour réaliser une transmission sécurisée d'information pour le transfert d'un droit d'utilisation de logiciel protégé du dispositif portatif (CL) vers un autre lecteur (LCL) ou à d'autres dispositifs portatifs (CL).

9. Système selon l'une quelconque des revendications précédentes caractérisé en ce qu'il comprend en périphérique de sauvegarde externe sécurisé contre les lecture et/ou les modifications non autorisées.

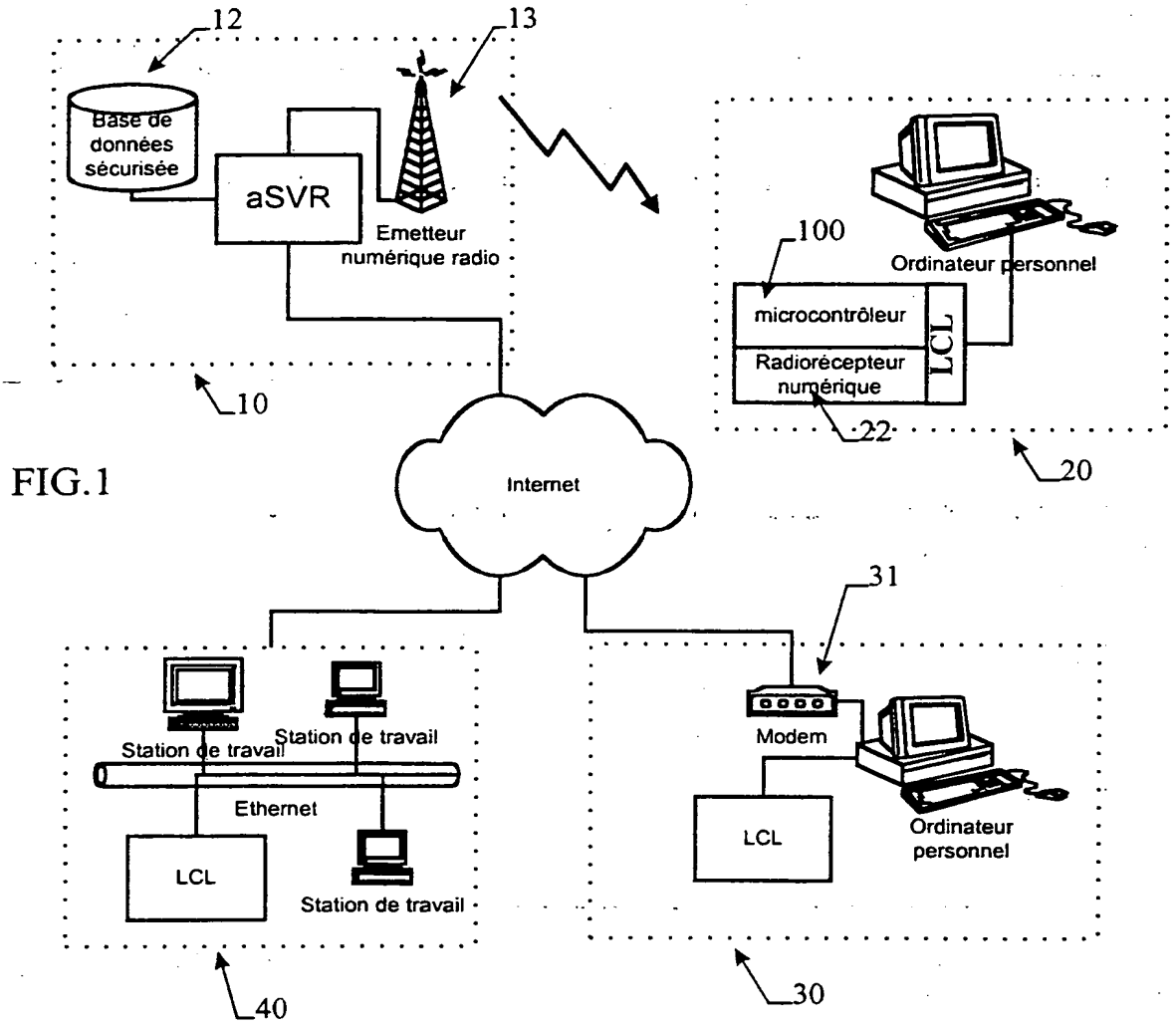


FIG.1

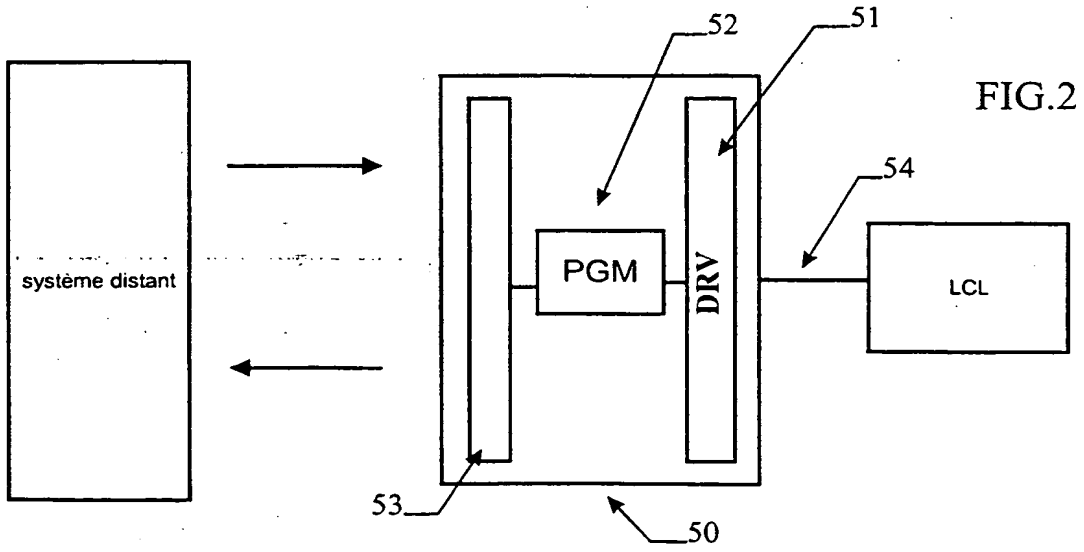
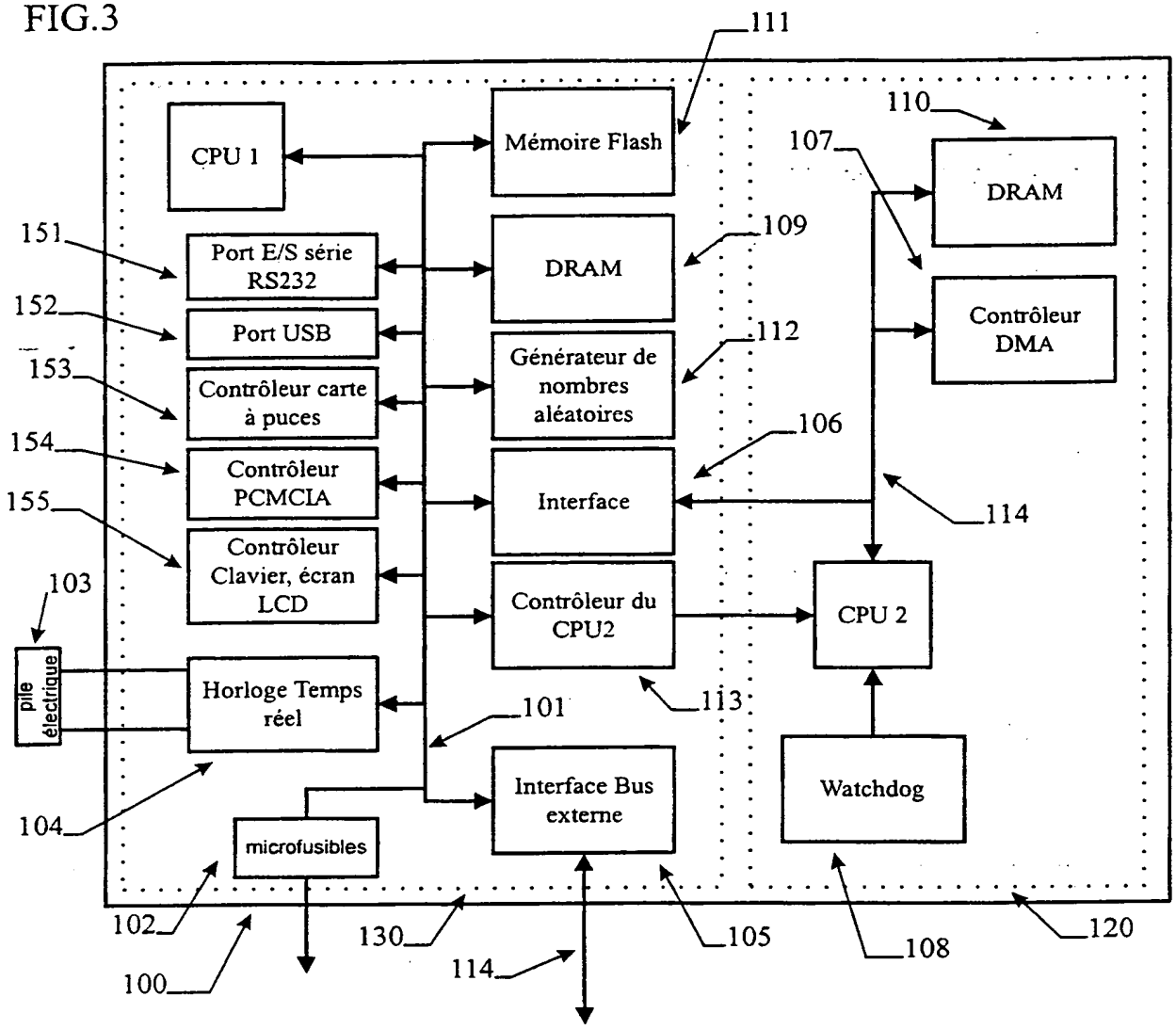


FIG.2

FIG.3



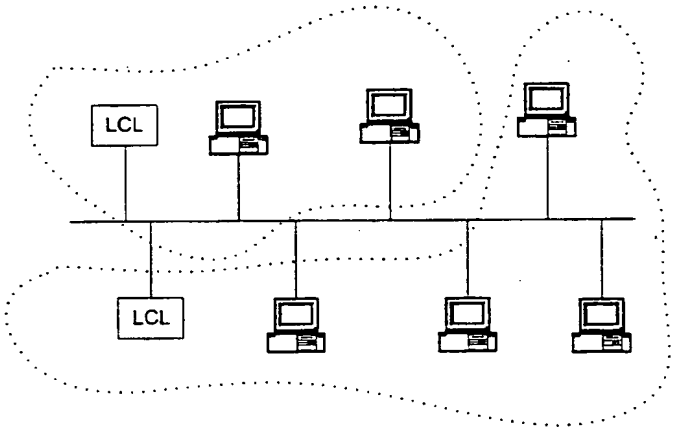


FIG.4

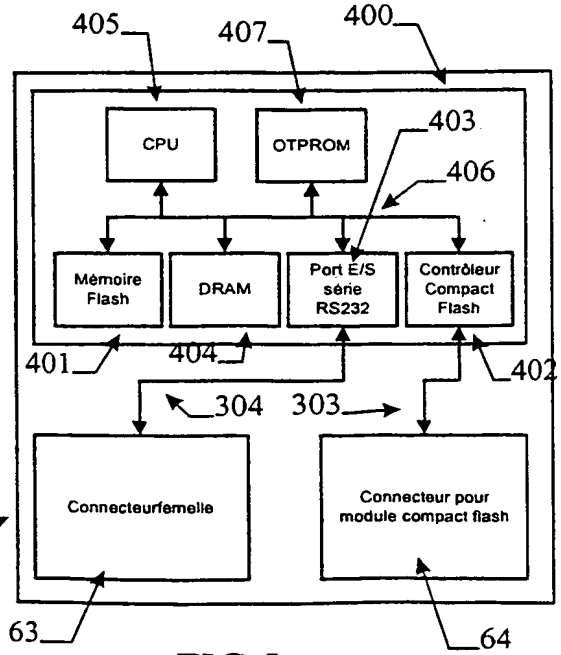


FIG.5

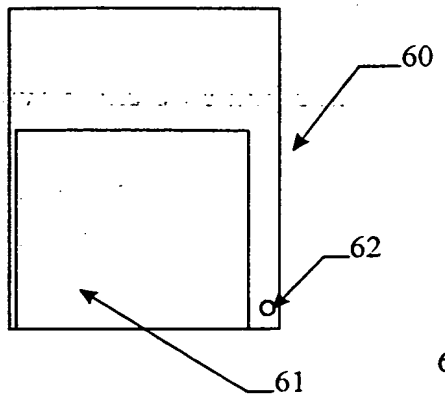


FIG.6

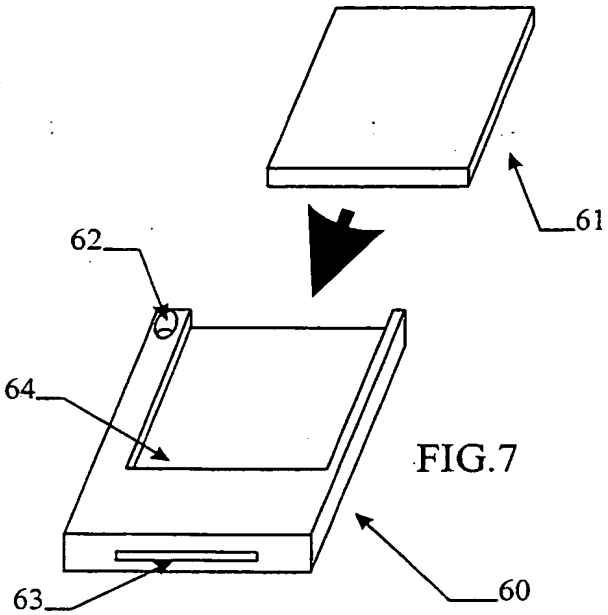


FIG.7

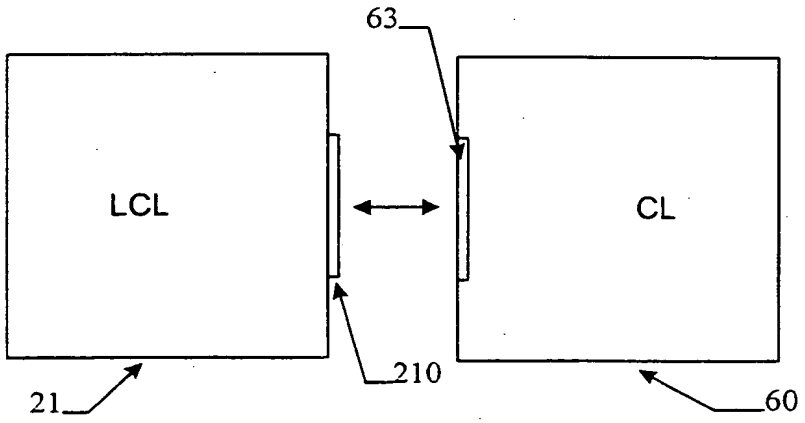


FIG.8

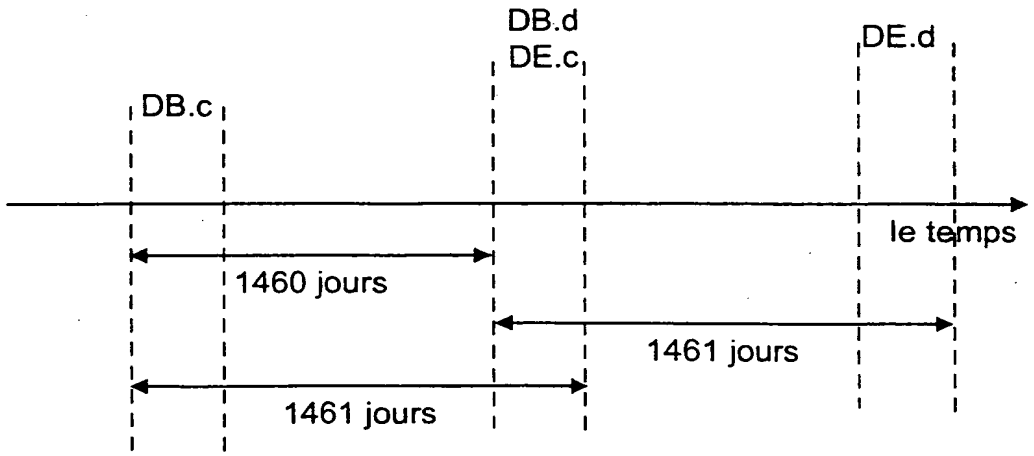


FIG.9

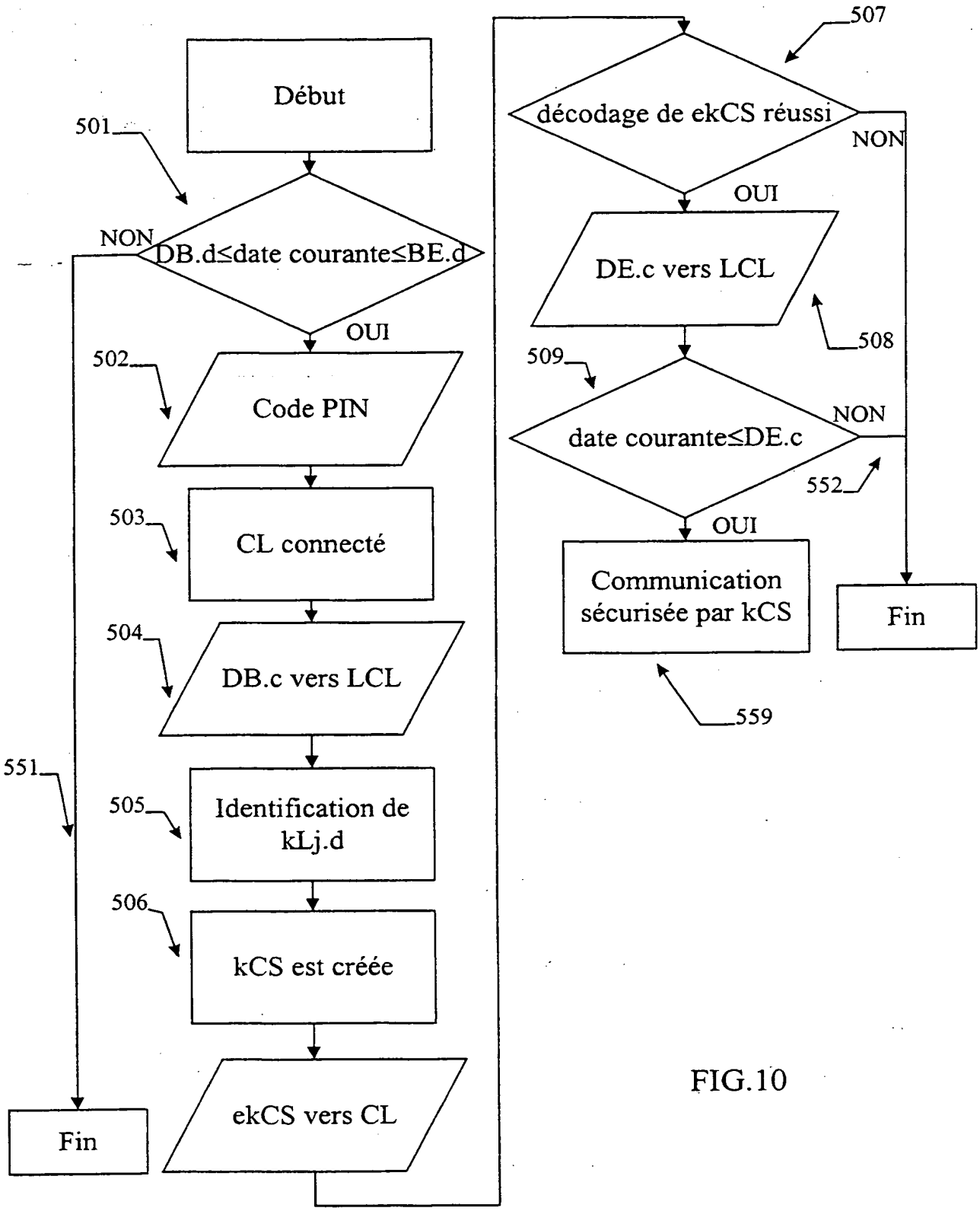


FIG.10

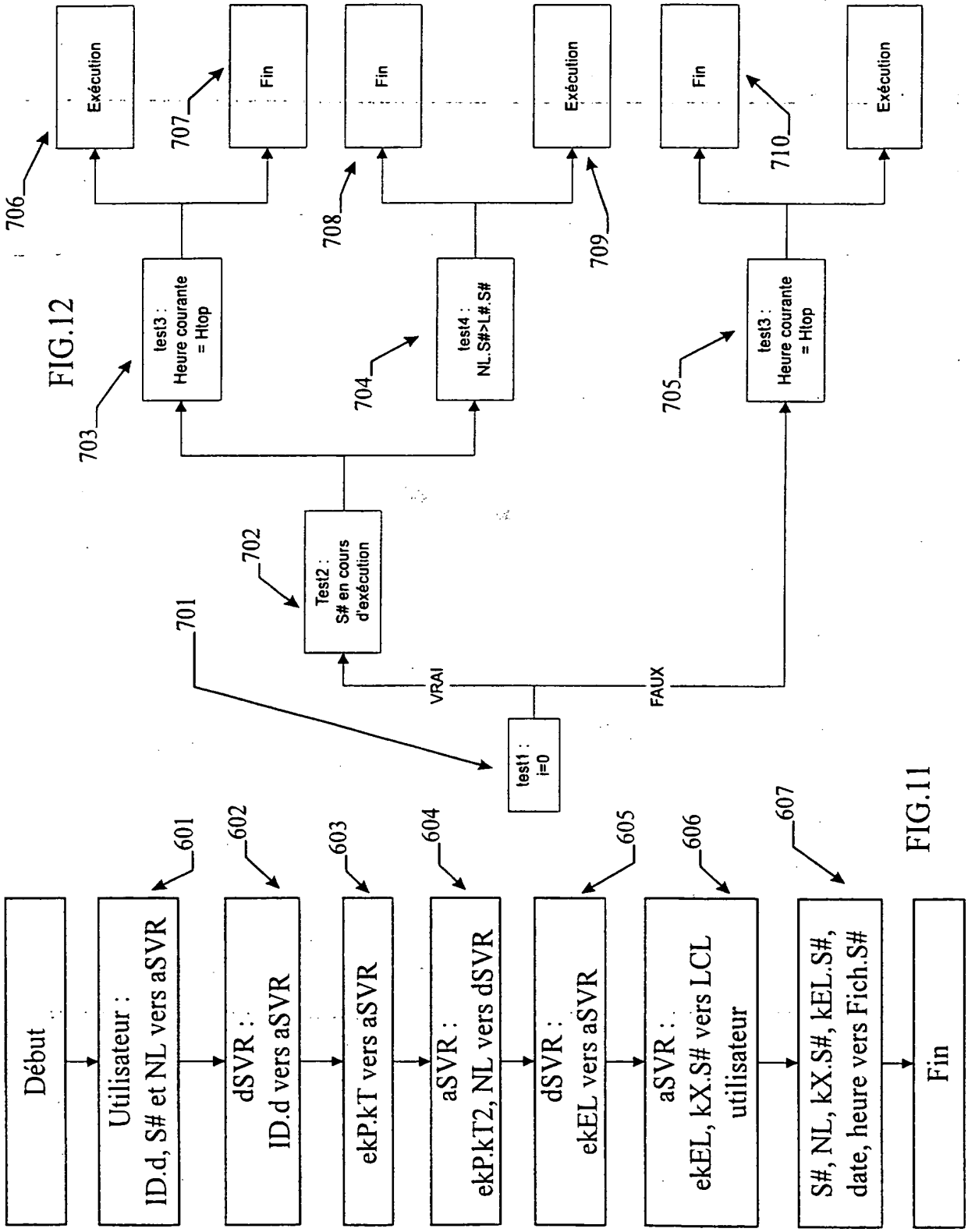


FIG.11

FIG.12

INTERNATIONAL SEARCH REPORT

Int« onal Application No
PCT/FR 99/00182

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 97 40448 A (CHOU ET AL.) 30 October 1997 see page 1, line 24 - line 35 see page 10, line 1 - line 21; figures 1A,1B	1,2,4,8, 9
A	EP 0 089 876 A (CII HONEYWELL BULL) 28 September 1983 see page 5, line 15 - page 7, line 22 see page 8, line 18 - page 9, line 11; figure	1,2,4,8, 9
A	WO 97 04412 A (CABLE TELEVISION LABORATORIES INC.) 6 February 1997 see page 6, line 15 - page 7, line 17; figure 1	1,4,6,7, 9

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 April 1999	06/05/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Taylor, P

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/FR 99/00182

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropnate, of the relevant passages	Relevant to claim No.
A	EP 0 613 073 A (INTERNATIONAL COMPUTERS LTD.) 31 August 1994 see column 2, line 17 - column 3, line 18 see column 5, line 14 - line 53; figure 1	1,2,8
A	US 5 155 680 A (WIEDEMER) 13 October 1992 see column 4, line 2 - line 63; figures 1-3	1,2,8
A	EP 0 457 677 A (TELEMECANIQUE) 21 November 1991 see column 4, line 31 - column 5, line 46; figure 1	1,7

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/FR 99/00182

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9740448 A	30-10-1997	US 5826011 A	20-10-1998
EP 89876 A	28-09-1983	FR 2523745 A	23-09-1983
		AT 27068 T	15-05-1987
		JP 58176746 A	17-10-1983
		US 4683553 A	28-07-1987
WO 9704412 A	06-02-1997	US 5754646 A	19-05-1998
		CA 2227060 A	06-02-1997
		GB 2317476 A	25-03-1998
EP 613073 A	31-08-1994	AU 667155 B	07-03-1996
		AU 5522894 A	01-09-1994
		ZA 9306234 A	21-03-1994
US 5155680 A	13-10-1992	US 4796181 A	03-01-1989
		CA 1281418 A	12-03-1991
		EP 0265183 A	27-04-1988
		JP 63191228 A	08-08-1988
		US 5047928 A	10-09-1991
EP 457677 A	21-11-1991	FR 2662280 A	22-11-1991
		CA 2042550 A	17-11-1991
		JP 4229345 A	18-08-1992

RAPPORT DE RECHERCHE INTERNATIONALE

Demande Internationale No
PCT/FR 99/00182

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 6 G06F1/00

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 6 G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	WO 97 40448 A (CHOU ET AL.) 30 octobre 1997 voir page 1, ligne 24 - ligne 35 voir page 10, ligne 1 - ligne 21; figures 1A,1B ---	1,2,4,8,9
A	EP 0 089 876 A (CII HONEYWELL BULL) 28 septembre 1983 voir page 5, ligne 15 - page 7, ligne 22 voir page 8, ligne 18 - page 9, ligne 11; figure ---	1,2,4,8,9
A	WO 97 04412 A (CABLE TELEVISION LABORATORIES INC.) 6 février 1997 voir page 6, ligne 15 - page 7, ligne 17; figure 1 --- -/--	1,4,6,7,9

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
27 avril 1999	06/05/1999

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Taylor, P

1

Formulaire PCT/ISA/210 (deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR 99/00182

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	EP 0 613 073 A (INTERNATIONAL COMPUTERS LTD.) 31 août 1994 voir colonne 2, ligne 17 - colonne 3, ligne 18 voir colonne 5, ligne 14 - ligne 53; figure 1	1,2,8
A	US 5 155 680 A (WIEDEMER) 13 octobre 1992 voir colonne 4, ligne 2 - ligne 63; figures 1-3	1,2,8
A	EP 0 457 677 A (TELEMECANIQUE) 21 novembre 1991 voir colonne 4, ligne 31 - colonne 5, ligne 46; figure 1	1,7

Formulaire PCT/ISA/210 (suite de la deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Den e Internationale No
PCT/FR 99/00182

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)		Date de publication
WO 9740448	A	30-10-1997	US 5826011	A	20-10-1998
EP 89876	A	28-09-1983	FR 2523745	A	23-09-1983
			AT 27068	T	15-05-1987
			JP 58176746	A	17-10-1983
			US 4683553	A	28-07-1987
WO 9704412	A	06-02-1997	US 5754646	A	19-05-1998
			CA 2227060	A	06-02-1997
			GB 2317476	A	25-03-1998
EP 613073	A	31-08-1994	AU 667155	B	07-03-1996
			AU 5522894	A	01-09-1994
			ZA 9306234	A	21-03-1994
US 5155680	A	13-10-1992	US 4796181	A	03-01-1989
			CA 1281418	A	12-03-1991
			EP 0265183	A	27-04-1988
			JP 63191228	A	08-08-1988
			US 5047928	A	10-09-1991
EP 457677	A	21-11-1991	FR 2662280	A	22-11-1991
			CA 2042550	A	17-11-1991
			JP 4229345	A	18-08-1992